SCALABLE ALGORITHMS FOR HIGH DIMENSIONAL STRUCTURED DATA

A Dissertation

by

SHAOGANG REN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Xiaoning Qian
Committee Members,	Edward Dougherty
	Jianhua Huang
	Peng Li
	Srinivas Shakkottai
Head of Department,	Miroslav Begovic

December 2017

Computer Engineering

ABSTRACT

Emerging technologies and digital devices provide us with increasingly large volume of data with respect to both the sample size and the number of features. To explore the benefits of massive data sets, scalable statistical models and machine learning algorithms are more and more important in different research disciplines. For robust and accurate prediction, prior knowledge regarding dependency structures within data needs to be formulated appropriately in these models. On the other hand, scalability and computation complexity of existing algorithms may not meet the needs to analyze massive high-dimensional data. This dissertation presents several novel methods to scale up sparse learning models to analyze massive data sets. We first present our novel **s**afe **a**ctive **i**ncremental **f**eature (SAIF) selection algorithm for LASSO (least absolute shrinkage and selection operator), with the time complexity analysis to show the advantages over state of the art existing methods. As SAIF is targeting general convex loss functions, it potentially can be extended to many learning models and big-data applications, and we show how support vector machines (SVM) can be scaled up based on the idea of SAIF. Secondly, we propose screening methods to generalized LASSO (GL), which specifically considers the dependency structure among features. We also propose a scalable feature selection method for non-parametric, non-linear models based on sparse structures and kernel methods. Theoretical analysis and experimental results in this dissertation show that model complexity can be significantly reduced with the sparsity and structure assumptions.

ACKNOWLEDGMENTS

First of all, I would like to express my deepest respect and gratitude to my advisor, Professor Xiaoning Qian, for his guidance and support over the years. Professor Qian's broad knowledge and research skills guided me to a quick start on the road of research and helped me to avoid many wrong paths. His support and guidance to my exploration in technical machine learning research lead to these important research topics. Without his persistent help and support, I would have never completed this dissertation.

I also would like to express my respect and gratitude to Professor Bo Zeng from the University of Pittsburgh for his guidance in optimization in the early stages of my doctoral training. His knowledge and attitude to technical research inspired me a lot. I would like to thank Professor Jianhua Huang, one of my committee members, for his kind help in the SAIF project, and I also learned a lot of inspiring ideas from his machine learning class.

I want to take this opportunity to thank the other committee members, Professor Edward Dougherty, Professor Peng Li, and Professor Srinivas Shakkottai for their time, interest, and suggestions. I'm also grateful to my friends Yijie Wang, Xingde Jiang, Chung-Chi Tsai, and Hyundoo Jeong. The technical communications with them also helped me a lot in my research.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Xiaoning Qian and Professors Edward Dougherty, Peng Li, and Srinivas Shakkottai of the Department of Electrical and Computer Engineering and Professor Jianhua Huang of the Department of Statistics.

Part of the data analyzed for Chapter 3 was provided by Professor Jieping Ye from University of Michigan. Part of the data analyzed for Chapter 4 was provided by Professor Xenios Papademetris from Yale University.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

This work was made possible in part by National Science Foundation under Grant 1553281 and Juvenile Diabetes Research Foundation under Grant 1-PNF-2014-151-A-V.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Massive data processing is becoming more and more important in modern research. To explore the benefits of massive data sets, scalable models have been studied by different research communities [1]. Prior knowledge regarding sparsity and structures within data sets are formulated as L_1 penalty and its variants to improve model robustness and prediction accuracy [2]. Deep models have been recently developed to model complicate data representations and structures and thus improve prediction accuracy [3].

On the other hand, the computation cost coming with these models on massive data sets is usually frightening. To tackle the problem, one way is to construct parallel or distributed systems and develop corresponding algorithms [1, 4]. This approach scales up well especially when the targeted problem can be paralleled. Stochastic gradient descent (SGD) [5, 6, 7] is another approach to combat problems with large data samples. SGD has been widely employed in non-convex complex problems such as training deep learning models. Besides distributed algorithms and SGD, recently people have developed methods that gain scalability relying on intrinsic sparse data structures. Problem size can be reduced by leveraging sparse structures recovered by the model. Feature screening methods such as [8, 9, 10, 11] can remove inactive or unimportant features to reduce the problem size and thus save CPU time in training. Sample screening methods such as [12, 13, 14, 15] provide or develop practicable approaches scaling support vector machines (SVMs) up for large data sets.

This dissertation proposes several methods for scaling up sparse models. In Chapter 2, 3 and 5, we develop approaches that can improve computation efficiency of sparse models along the screening strategy. In Chapter 4, we present a scalable structured kernel feature selection method that can be scaled up with dual average stochastic approximation

algorithm. Chapter 2 and 3 deal with data sets with large feature size, and Chapter 4 and 5 are for data sets with large sample size.

There are three sections in this introduction chapter. Some basics on least absolute shrinkage and selection operator (LASSO), support vector machine (SVM), kernel feature selection, and screening methods are given in the first section. Research motivations are given in the second section. The third section summarizes the main contributions of this dissertation.

1.1 Mathematical Background

In this section we first survey the basic concepts such as LASSO, SVM, and kernel feature selection for which we will provide efficient algorithms in following chapters. All of these models are sparse models in which part of the optimal model parameters could be zero. LASSO and kernel feature selection relies on L_1 norm to obtain sparsity, while SVM is a non-parametric model that can automatically assign zero coefficients to non-support samples adaptively based on training data complexity. Literature reviews on feature and sample screening are given in the last subsection.

1.1.1 LASSO

Least **a**bsolute **s**hrinkage and **s**election **o**perator (LASSO) and its variations have been wildly used for feature selection, sparse structure recovering, compressed sensing and so on. Let $X \in \mathcal{R}^{n\times p}$ be a data matrix with n samples and p features, and $\mathbf{y} \in R^{p\times 1}$ is the response vector. The original LASSO problem [16] is as follows:

$$\min_{\beta} \frac{1}{2}||\mathbf{y} - X\beta||_2^2 + \lambda||\beta||_1. \tag{1.1}$$

Here λ is the regularization parameter. The L_1 penalty term imposes sparsity on β, and this leads to some entries of the optimal solution β^* being zeros. One variant of LASSO

is Fused LASSO [17],

$$\min_{\beta} \frac{1}{2}||\mathbf{y} - X\beta||_2^2 + \lambda \sum_{i=1}^{p-1} |\beta_i - \beta_{i+1}| \ . \tag{1.2}$$

From the formulation, Fused LASSO tries to make adjacent model variables to be the same, and this corresponds to the chain structures within many data sets such as time series data. The Fused lasso can be rewritten as

$$\min_{\beta} \frac{1}{2}||\mathbf{y} - X\beta||_2^2 + \lambda||D\beta||_1 \ , \tag{1.3}$$

where

$$D = \begin{bmatrix} 1 & -1 & 0 & ... & 0 & 0 \\ 0 & 1 & -1 & ... & 0 & 0 \\ & & & ... & & \\ 0 & 0 & 0 & ... & 1 & -1 \end{bmatrix} . \tag{1.4}$$

Fused LASSO and the matrix form (1.3) can be extended to a broader range of tree and graph structures, and all of these are named generalized LASSO that we will present a novel scaling up method in the second chapter.

A bunch of algorithms have been brought up to solve the LASSO problem, such as shooting algorithm [18], basis pursuit method [19], grafting [20], etc.. Feature screening methods [10, 11, 21] have been developed to scale up LASSO, and we will give a detailed review on these approaches.

1.1.2 Support Vector Machine

Suppose we have a dataset $\mathcal{D} = \{(x_i, y_i)\}_n$, and $x_i \in R^d$, $y_i \in \{-1, 1\}$. Let ψ be feature mapping function, $\psi : \mathcal{X} \to \mathcal{F}$. Let w be a vector in feature space $\mathcal{F}$, the primal

problem for SVM is:

$$P: \min_{w} \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{n}[1 - w^T(y_i\psi(x_i))]_+ \tag{1.5}$$

Here $C \in R^+$ is the model penalty parameter, and a small C corresponds wide decision margin. And the corresponding dual problem [14, 12, 22, 15] is

$$\hat{D}: \sup_{\theta} -\frac{1}{2}||Z^T\theta||_2^2 + 1^T\theta \tag{1.6}$$

$$s.t. \qquad \theta_i \in [0, C], \ \ \forall i, \tag{1.7}$$

where

$$Z = [y_1\psi(x_1), y_2\psi(x_2), ..., y_l\psi(x_n)]^T. \tag{1.8}$$

Let w^* and θ^* denote the optimal solution to primal and dual problem. We have the primal and dual relationship as

$$w^* = Z^T\theta^*. \tag{1.9}$$

If we use $Q = ZZ^T$, the dual problem is a standard quadratic optimization problem:

$$D: \min_{\theta} \frac{1}{2}\theta^T Q\theta - 1^T\theta \tag{1.10}$$

$$s.t. \qquad \theta_i \in [0, C], \ \ \forall i. \tag{1.11}$$

Many algorithms have been developed to address training SVM. Coordinate descent methods have been developed for linear SVM [23, 24]. Sequential minimal optimization (SMO) methods [25, 26] can solve large scale kernel SVM by searching the well chosen

directions. Stochastic gradient decent methods have been extended to SVM on large data sets in [27, 28]. Similar to LASSO, sample screening methods have been proposed to solve SVM on large data sets.

1.1.3 Kernel Feature Selection

People have brought up non-linear feature selection models to capture intrinsic response relationship between variables. Kernel feature selection is an import type of non-linear feature selection method. For example, the formulation for the Hilbert-Schmidt Feature Selection (HSFS) [29] is as follows:

$$\min_{W \in R^{P \times P}} -HSIC(WX, Y) + \lambda \sum_{i=1}^{P} ||w_i||_\infty, \tag{1.12}$$

where $W = [w_1, ..., w_d]$ is a transformation matrix. Limited-memory BFGS (L-BFGS) algorithm [30] can be used to solve the problem. One limitation of HSFS is that the objective function is non-convex. Hence, with different starting points for optimization, we may get different solutions. Other kernel based feature selection methods include HSIC, FVM, HSIC-LASSO [31, 32, 33]. In [31], they propose to minimize the following objective function:

$$\min_{\alpha} \frac{1}{2}||\bar{L} - \sum_{k=1}^{p} \alpha_k \bar{K}|| + \lambda ||\alpha||_1 \tag{1.13}$$

$$s.t. \quad \alpha_k \geq 0, \ \forall k = 1, ..., p \ . \tag{1.14}$$

The loss function can be interpreted as

$$\frac{1}{2}||\bar{L} - \sum_{k=1}^{p} \bar{K}|| = \frac{1}{2} HSIC(Y,Y) - \sum_{i} a_i HSIC(Y, X_{\bullet i}) \tag{1.15}$$

$$+ \frac{1}{2} \sum_{ij} a_i a_j HSIC(X_{\bullet i}, X_{\bullet j}). \tag{1.16}$$

With the last term, their methods aim to eliminate the correlated redundant features. We will propose a novel structured kernel feature selection model in chapter 4.

1.1.4 Screening Method for Sparse Models

In this subsection, we review the screening methods developed recently by researchers for sparse models such as LASSO and SVM.

1.1.4.1 Sequential Feature Screening

Traditional methods such as shooting algorithm(coordinate minimization with soft-thresholding) have been proposed to solve the LASSO problems. However, with large p and n, this type of problem will become difficult to solve. Recently feature screening has been proposed to scale up sparse learning. The first type of feature screening method is sequential screening. Most sequential screening methods derive screening rules by leveraging the solutions to the LASSO model with a heavier regularization parameter.

There are two broad categories of sequential screening methods for LASSO problems: heuristic and safe screening methods. The heuristic screening methods [8, 9] relies on heuristics to remove features. For example, the Strong Rule screening [8] derives the screening rule based on the assumption that the absolute values of the inner products between features and the residue are non-expansive with respect to the parameter values. It is obvious that this assumption does not always hold. Such heurstic screening rules are not safe, meaning that they cannot guarantee that the removed features will have corresponding zero value in the optimal LASSO solution to the original full-scale problem.

Safe LASSO screening methods do not take any unsafe assumption that the heuristic screening methods use. Most of the safe screening methods [10, 11, 21] are inspired by the seminal work by [10] and derive screening rules with the help of the LASSO solution with a heavier regularization parameter. To derive the screening rule, we first need to have the dual form of LASSO problem (1.1), which is given by

$$\sup_{\theta} \frac{1}{2}||\mathbf{y}||_2^2 - \frac{\lambda^2}{2}||\theta - \frac{\mathbf{y}}{\lambda}||_2^2 \tag{1.17}$$

$$s.t. \quad |x_i^T\theta| \leq 1, \forall i = 1, ..., p\ . \tag{1.18}$$

(1.17) is a strongly convex quadratic problem with polygon constrains. For problem (1.1), with KKT conditions [34, 11], we have

$$x_i^T\theta^* \in \begin{cases} sign([\beta^*]_i) & if [\beta^*]_i \neq 0 \\ [-1, 1] & if [\beta^*]_i = 0 \end{cases}. \tag{1.19}$$

The primal and dual variable relationship is $\mathbf{y} - X\beta = \lambda\theta_j$. From (1.19), we have

$$|x_i^T\theta^*| < 1 \implies [\beta^*]_i = 0 \implies x_i \text{ inactive feature.}$$

Given the optimal dual variables θ^*, we can easily check whether feature i is active or not by $|x_i^T\theta^*| < 1$. As it is equally expensive to compute θ^* compared to solving the original LASSO problem, screening methods aim to estimate a convex or ball region $B(\theta, r) = \{\theta^* \mid ||\theta^* - \theta||_2 \leq r\}$ as the range of θ^*. With $\theta^* \in B(\theta, r)$, let $\theta^* = \theta + \rho$, we can see $||\rho||_2 \leq r$. With $x_i^T\theta^* = x_i^T\theta + x_i^T\rho$, we have

$$x_i^T\theta - ||x_i||_2 r \leq x_i^T\theta^* \leq x_i^T\theta + ||x_i||_2 r.$$

Then

$$\text{if } \begin{cases} x_i^T\theta - ||x_i||_2 r > -1 \\ x_i^T\theta + ||x_i||_2 r < 1 \end{cases} \implies x_i \text{ inactive feature.} \tag{1.20}$$

Clearly, the tightness of the bound estimates and the computational cost of deriving $B(\theta, r)$ determine the effectiveness of the corresponding screening methods.

Sequential screening methods rely on the LASSO solution with a heavier penalty to infer the ball region of the dual variables θ, $B(\hat{\theta}^*(\lambda'), r)$. Here $\lambda' > \lambda$, and $\hat{\theta}^*(\lambda')$ can be computed based on the primal-dual relation when the solution to the LASSO problem with λ' as the regularization penalty parameter. For example, DDP [11] takes the dual problem (1.17) as a projection problem and estimates the ball range of θ^* based on the properties of projection operators such as non-expansiveness. Based on DPP, the screening rules for Group LASSO [21], 1D-chain Fused LASSO [35], Sparse Group LASSO [36], and Tree Group LASSO [37] have been developed. Typically, sequential screening requires to solve a sequence of LASSO problems corresponding to a sequence of descending λ's to gradually tighten the range estimates of θ^* to achieve the high screening power.

1.1.4.2 Dynamic Feature Screening

Instead of relying on the solutions with different λ's, the recently proposed dynamic screening [38, 39, 40] directly derives the range estimates of θ^* by strong duality based on the strong convex property of the dual objective function. The ball region for θ^* is estimated based on the duality gap as a function of the primal and dual objective function values at iterative updates [38, 39]:

$$\forall \theta \in \Omega_{\mathcal{F}}, \beta \in R^{p\times 1}, \;\; B\Big(\theta, \frac{2}{\lambda^2}[P(\beta) - D(\theta)]\Big) = \Big\{\theta^* \mid ||\theta^* - \theta||_2^2 \leq \frac{2}{\lambda^2}\big[P(\beta) - D(\theta)\big]\Big\}. \tag{1.21}$$

Here $\Omega_{\mathcal{F}}$ is the dual feasible space corresponding to feature set $\mathcal{F}$. β is the current estimation of primal variable, and θ is the projected feasible dual variable of β. The tightness of the results depends on the duality gap $[P(\beta) - D(\theta)]$, determined by the quality of iterative updates β and θ. Dynamic screening algorithms in [38, 39] iteratively update β and θ for the original LASSO problem with the whole feature set X to check the duality gap and apply screen rules to remove inactive features. Without the solution information from a heavier parameter, dynamic screening has to iterate the operations in optimization, such as sub-gradient computation, on the original whole feature set many times to gain a small duality gap. Within these iterations, a large number of redundant solf-threshold or sub-gradient operations can be performed on inactive features.

1.1.4.3 Sample Screening for SVM

Support Vector Machines gain their sparse structures on support vectors. Similar to sequential screening for LASSO, sample screening method [12, 13] derive their screening rules by leveraging the solutions to SVM with a another hyper parameter.This type of sample screening methods have been extended to sparse SVM in [41]. Recently, the screening method developed in [15] derives sample screening rules by leveraging the duality gap, which is similar to the dynamic screening method for sparse learning [38].

Most of SVM sample screening rules are derived based on the dual form (1.10). With KKT condition regarding to (1.10), we have

$$[\theta^*(C)_i] = 0, \text{ if } \langle Z^T\theta, y_i\psi(x_i)\rangle - 1 > 0 \tag{1.22}$$

$$[\theta^*(C)_i] = C, \text{ if } \langle Z^T\theta, y_i\psi(x_i)\rangle - 1 < 0 \tag{1.23}$$

$$[\theta^*(C)_i] \in [0, C], \text{ if } \langle Z^T\theta, y_i\psi(x_i)\rangle - 1 = 0 \tag{1.24}$$

Let us use $\{ \mathcal{R}, \mathcal{L}, \mathcal{E} \}$ to represent the sets of data samples satisfy the three conditions.

$$\mathcal{R} := \{i \in \mathbb{N} | \langle Z^T\theta, y_i\psi(x_i)\rangle > 1\} \tag{1.25}$$

$$\mathcal{L} := \{i \in \mathbb{N} | \langle Z^T\theta, y_i\psi(x_i)\rangle < 1\} \tag{1.26}$$

$$\mathcal{E} := \{i \in \mathbb{N} | \langle Z^T\theta, y_i\psi(x_i)\rangle = 1\} \tag{1.27}$$

The data instance in $\mathcal{E}$ are on margin support vectors, data instances in $\mathcal{L}$ are inside margin support vectors, and data instances in $\mathcal{R}$ are non-support vectors. With the solution from C', [12, 13] estimate a region for either θ or $\mathbf{w}$, and then remove a subset of the non-support vector samples. Reduced sample size leads to less CPU time and memory space.

1.2 Motivations

LASSO and its variants are powerful tools for feature selection. On one hand the L_1 norm can recover sparsity structures in data, while on the other hand its non-smoothness results in difficulties in optimization. For sparse models, when the data set is with large feature or sample size, the computation cost will become one of the main factors people need to consider. As mentioned in previous section, screening methods provide us approaches that can avoid redundant computation resulted from inactive features.

There are drawbacks coming with sequential and dynamic screening methods for LASSO. Sequential screening relies on the model solution with a heavier penalty to infer the ball region of the dual variables. The closer two λ values are, the tighter the range estimates of θ^* can be. Such a sequential procedure is suitable and efficient when solving a sequence of such problems with different regularization parameters is necessary, for example, for model hyper-parameter selection by cross validation. However, in the situations where we only want to derive the solution with a small number of specific λ values, sequential screening may take too much redundant computation on irrelative λ values. For dynamic

screening, as mentioned in previous section, we may need many iterations to reach the duality gap with screening power. The computation cost of the redundant operations on inactive features or samples dilute the screening benefits.

Another shortcoming for existing screening methods is that they do not consider general variable dependence structures, such as graph structure presented in generalized LASSO (GL). With a generic structure in $|D|$ in (1.3), the dual form will become much more complicate, and it is not easy to derive sequential screening rules by following the strategies utilized for LASSO and group LASSO. Thus we do need new screening strategies for GL problems.

Furthermore, all of these screening methods are targeting at linear models with large feature size. While it is equally challenging to solve non-linear feature selection models such as kernel feature selection with large sample size. As described in previous section, large sample size could result in infeasible kernel feature selection models. Thus there is large scaling up space for kernel feature selection models.

In this dissertation, we try to bring up several methods to tackle these challenging problems. We list the main contributions in next section.

1.3 Main Contributions

We propose several techniques that can further boost structured sparse models. We summary our contributions in each chapter as follows.

Chapter 2 presents the important contributions of this dissertation, scalable safe active incremental feature selection (SAIF). SAIF can overcome the shortcomings of sequential and dynamic screening, and scales up sparse models such as LASSO by maximumly reducing the redundant computation resulted from inactive features. Starting from an empty active feature set, SAIF dynamically recruits the most correlated features and removes inactive features with the estimation of the dual variables. Experimental results show sig-

nificant improvements over existing screening methods. Theoretical analysis also proves the advantages of SAIF.

In Chapter 3, we focus on screening methods for generalized LASSO (GL). By leveraging the dual form of GL, we show that GL screening rules rely on efficient deriving the bounds of the solution space of an inequality system. We present an efficient approximation approach to tackle this problem. We also show how to extend SAIF to tree Fused LASSO, a special case of GL. Experiments on simulation and real-world data sets demonstrate the advantages of our methods. The proposed methods has broad applications and impacts as they applicable to general loss functions and variable dependency structures.

In Chapter 4, we discuss a novel scalable method for kernel feature selection with structures. The proposed model can incorporate general graph structures, such as 2D and 3D image grid, into kernel feature selection. The model formulation comes with the advantages that it can easily be scaled up with the dual average stochastic gradient descent method [42]. Results from 3D image analysis show that the proposed model not only obtains improved accuracy but also can save computation time tremendously.

Chapter 5 extends the idea of SAIF to SVM and leads to scalable safe active incremental support vector selection (SAIV) algorithm. Support vectors give SVM models sparsity, and this also provide them the scaling up opportunities by leveraging the idea of SAIF. Experiments and theoretical results are presented to demonstrate that SAIV can reduce the computation cost of training SVM models.

The proposed models and methods, in which sparsity and structures can be incorporated as prior knowledge, can boost prediction performance and improve model efficiency as well.

2. SAFE ACTIVE FEATURE SELECTION FOR SPARSE LEARNING

In this chapter, we describe a novel method to scale up LASSO solutions, **s**afe **a**ctive **i**ncremental **f**eature selection (SAIF). SAIF is different from the existing sequential screening and dynamic screening methods for LASSO, both of which require solving the full-scale LASSO problem in the original feature space. SAIF does not require a solution from a heavier penalty parameter as in sequential screening or update the full model for each iteration as in dynamic screening. SAIF starts with a small number of features and only updates the significantly reduced model with the current most active features. The iterative procedure of SAIF incrementally recruits active features and updates the model to reach the final LASSO solution with the convergence guarantee to achieve the optimal solution to the original full LASSO problem. SAIF has a promising potential to solve the scalability issue for LASSO and its extensions when facing extremely high dimensional data sets. Experiments with both synthetic and real-world data sets show that SAIF can be up to 50 times faster than dynamic screening, and hundreds of times faster than LASSO solutions without screening.

2.1 Introduction

LASSO has been a powerful tool for sparse learning to generalize predictions based on analyzing data sets with $p \gg n$, where p is the number of covariates or features and n the number of samples. LASSO screening methods provide efficient approaches to scale up sparse learning without solving the full LASSO problems, based on either sequential or dynamic screening methods [10, 11, 38]. However, the existing sequential screening requires the LASSO solution with a heavier regularization penalty parameter so that the range of dual variables can be estimated tightly to help effectively screening redundant features. Different from such static sequential screening methods, dynamic screening does

not require the solution with the heavier penalty parameter but relies on duality gaps for feature screening. To achieve high screening power, a significant number of optimization iterations have to be operated on the full-scale problems with the original high dimensional feature set to compute the effective duality gap. Both sequential and dynamic screening requires to update the original full-scale LASSO model.

Homotopy methods have been applied to LASSO to compute the solution path when λ varies [43, 44, 45, 46, 47, 48]. This type of methods rely on a sequence of decreasing λ values and "warm start" (starting the active set with the solution from the previous λ) to achieve computational efficiency. Usually these methods have multiple iteration loops to incorporate the strong rule screening, active set, and path-wise coordinate descent. The inner loop performs coordinate descent and active set management. The outer loop goes through a sequence of decreasing λ values and initializes the active set at each λ with the strong rule and warm start. Since they do not utilize safe convergence stopping criteria for the active set, they may miss some of the optimal active features. Furthermore, this type of methods do not employ any screening rule for the inner-loop sub-problem, and it may limit the scalability.

Besides screening and homotopy methods, working set methods [49] maintain a working set according to some violation rules and solve a sub-problem regarding the working set at each step. The working set method [49] estimates an extreme feasible point based on the current solution, and then the constraints that are closest to the feasible point construct the working set for the next step. This kind of methods also start from solving the original full-scale problem as the existing LASSO screening methods. However, when $p \gg n$, the basic assumption of sparse learning is that most of the given features are irrelevant and should be inactive for the optimal solutions. It is clear that existing algorithms may not be efficient due to redundant time-consuming operations on inactive features. In this chapter, we propose a novel LASSO feature selection method to fur-

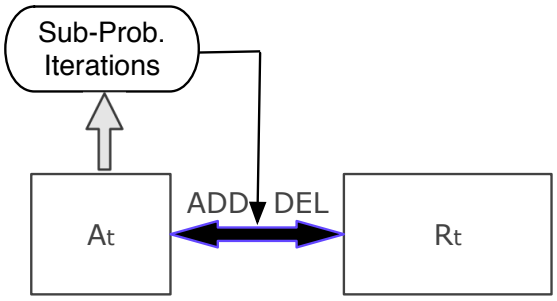


Figure 2.1: SAIF Screening. A_t stands for the Active set, while R_t stands for the Remaining set at step t.

ther scale up LASSO solutions by overcoming the issues in the existing methods. Rather than taking the whole feature set as the initial input, our method SAIF starts from a small set of features, which is taken as the active set. The features that are not in the active set are put in the remaining set (Figure 2.1). Time-consuming iterations such as coordinate minimization with soft-thresholding are only performed on the features in the active set. Features are actively recruited or removed from the active set according to the estimated ranges of optimal dual variables. Based on duality properties, efficient feature operation rules and safe stopping criteria have been developed to keep most inactive and redundant features out of the active set. With a small active set, CPU time and memory operations can be tremendously reduced. Complexity analysis is provided for both dynamic screening and SAIF. Theoretical results show that the running time of SAIF is almost only proportional to the active feature size, the number of features with non-zero model coefficients in the optimal LASSO, rather than the input feature size. Experiments on simulated and real-world datasets verified the advantages of the proposed method.

Data: Data matrix X, label Y, penalty λ, stopping duality gap ϵ

Result: Coefficient Vector β

Choose $\lceil c \log(\frac{md+mx}{\lambda}) \log(p) \rceil$ features from $\mathcal{F}$ in the descending order of $|X^T \mathbf{f}'(\mathbf{0})|$;

$\delta = \frac{\lambda}{\lambda_{max}}$, IsAdd = True;

while *True* **do**

  Update β_t with K iterations of soft-thresholding operations on $\mathcal{A}_t$;

  Compute a ball region $B(\theta_t, r_t)$ based on (2.13) or (2.14);

  $r_t = \delta r_t$;

  if *IsAdd = False & Duality Gap* $< \epsilon$ **then**
    Stop;

  end

  DEL operation;

  if *IsAdd = False* **then**
    Continue;

  else

    if $\max_{i \in R_t} |x_i^T \theta_t| + ||x_i||_2 r_t < 1$ **then**

      if $\delta < 1$ **then**
        $\delta = \min(10\delta, 1)$

      else
        IsAdd = False; Continue;

      end

    end

    ADD operation;

  end

end

Put β_t in to β, and inflate the other entries with 0.

Algorithm 1: SAIF Algorithm

2.2 Safe Active Incremental Feature Selection for LASSO

We derive an innovative incremental feature screening algorithm, SAIF, in which we can iteratively solve much smaller sub-problems, i.e., iteratively update the duality gap while adding or removing features by leveraging the active ball region estimates for the optimal dual variables of these sub-problems. The schematic illustration of SAIF is given in Figure 2.1. Let $\mathcal{A}_t$ and $\mathcal{R}_t$ denote the active feature index set and remaining feature index set at iteration step t, respectively. Instead of solving either the original full-scale LASSO primal problem or the corresponding dual problem, SAIF screening is different from the existing sequential and dynamic screening as it only needs to solve significantly reduced sub-problems and updates the screening rules only based on the duality gap without solving these sub-problems exactly. More importantly, SAIF has the safe guarantee that only irrelevant or redundant features in the original LASSO problem will be removed. Algorithm 1 summarizes our SAIF screening procedure, which starts with $\mathcal{A}_0$ and dynamically moves active features between $\mathcal{R}_t$ and $\mathcal{A}_t$.

2.2.1 ADD and DEL Operations

Two operations in SAIF are ADD and DEL. Starting from an initial active set $\mathcal{A}_0$, whose features can be selected by some simple heuristics, for example, based on their correlation with the output, SAIF iteratively adds features (ADD) into or removes features (DEL) from the active set. At the tth iteration, we derive both ADD and DEL operations to dynamically update $\mathcal{A}_t$ based on the primal sub-problem with only the current active

features:

$$P_t: \quad \min_{\beta \in R^{|\mathcal{A}_t| \times 1}} \sum_{i=1}^{n} f(\sum_{j:j \in \mathcal{A}_t} x_{ij}\beta_j, y_i) + \lambda||\beta||_1. \tag{2.1}$$

$$D_t: \quad \sup_{\theta} -\sum_{j=1}^{n} f^*(-\lambda\theta_j, y_j) \tag{2.2}$$

$$s.t. \quad |x_i^T\theta| \leq 1, \quad \forall i \in \mathcal{A}_t,$$

Let $\Omega_{\mathcal{A}_t}$ be the dual feasible region and $D(\theta_t)$ denote the dual objective function value of the sub-problem at the dual variable θ_t considering only the active features in $\mathcal{A}_t$ with θ_t^* being the corresponding optimal dual solution. We use $\beta_t \in R^{|\mathcal{A}_t| \times 1}$ to represent the β value after t out layer iterations in SAIF. β_t^* denotes the optimal active feature solution regarding the problem P_t. $P_t(\tilde{\beta})$ is the objective value of P_t with input $\tilde{\beta}$, and $\tilde{\beta}$ can have a different set of features compared with $\mathcal{A}_t$; we inflate the missing entries in $\tilde{\beta}$ with zeros and ignore the entries or features not in $\mathcal{A}_t$ in the calculation of $P_t(\tilde{\beta})$. Let $S_{\mathcal{A}}$ represent the set of the optimal primal solutions for any feature set $\mathcal{A}$, θ^* the optimal dual solution with the full feature set $\mathcal{F}$, and $\bar{\mathcal{A}}$ for the optimal active feature set that $\{i : |x_i^T\theta^*| = 1\}$. Let $B(\theta_t, r_t) = \{\theta_t^* \big| \, ||\theta_t^* - \theta_t||_2 \leq r_t\}$ be an estimated ball region for θ_t^* at step t.

SAIF carries out ADD and DEL operations as follows:

DEL: For $i \in \mathcal{A}_t$, if $|x_i^T\theta_t| + ||x_i||_2 r_t < 1$, move i from $\mathcal{A}_t$ to $\mathcal{R}_t$.

ADD: For $j \in \mathcal{R}_t$, if $\forall k \in \mathcal{R}_t, k \neq j$, $\big||x_j^T\theta_t| - ||x_j||_2 r_t\big| > |x_k^T\theta_t| + ||x_k||_2 r_t$, move j to $\mathcal{A}_t$.

We have the following theorem regarding ADD and DEL operations:

Theorem 1 *Assume $B(\theta_t, r_t) = \{\theta_t^* \big| \, ||\theta_t^* - \theta_t||_2 \leq r_t\}$, an estimated ball region for θ_t^* at step t.*

a) If we add a new feature into $\mathcal{A}_t$, then $\mathcal{A}_t \subseteq \mathcal{A}_{t+1}$, $\Omega_{\mathcal{A}_t} \supseteq \Omega_{\mathcal{A}_{t+1}}$, and $D(\theta_{t+1}^*) \leq D(\theta_t^*)$.

b) If $\exists i \in \mathcal{R}_t$ *and* $|x_i^T\theta_t^*| > 1$*, we add feature* i *to* $\mathcal{A}_t$ *at step* t*, then* $D(\theta_t^*) > D(\theta_{t+1}^*)$*.*

c) At step t*, if* $\max_{i\in\mathcal{R}_t} |x_i^T\theta_t^*| < 1$*, then* $\theta_t^* = \theta^*$*,* $\beta_t^* \in S_{\mathcal{F}}$*.*

d) If x_i *satisfies* $\forall j \in \mathcal{R}_t, j \neq i, \left||x_i^T\theta_t| - ||x_i||_2 r_t\right| \geq |x_j^T\theta_t| + ||x_j||_2 r_t$*, then* $|x_i^T\theta_t^*| \geq |x_j^T\theta_t^*|, \forall j \in \mathcal{R}_t, j \neq i$*.*

Proof: a) From the dual form (2.2), if we add i to $\mathcal{A}_t$, there will be one more constraint for the dual problem at step $t+1$, thus $\Omega_{\mathcal{A}_{t+1}} \subseteq \Omega_{\mathcal{A}_t}$. As we have smaller feasible space at $t+1$, $D(\theta_{t+1}^*) \leq D(\theta_t^*)$.

b) As $\Omega_{\mathcal{A}_{t+1}} \subset \Omega_{\mathcal{A}_t}$, we have $D(\theta_{t+1}^*) \leq D(\theta_t^*)$. With $|x_i^T\theta_t^*| > 1$ and $|x_i^T\theta_{t+1}^*| \leq 1$, $\theta_{t+1}^* \neq \theta_t^*$. As $\Omega_{\mathcal{A}_t}$ is convex and closed, and $\mathbf{f}^*$ is convex and smooth, the optimal dual solution for the active set $\mathcal{A}_t$ is unique, which means $D(\theta_t^*) \neq D(\theta_{t+1}^*)$. Hence, $D(\theta_t^*) > D(\theta_{t+1}^*)$.

c) According to a), with $\mathcal{A}_t \subseteq \mathcal{F}$, we have $\Omega_{\mathcal{F}} \subseteq \Omega_{\mathcal{A}_t}$, and $D(\theta^*) \leq D(\theta_t^*)$. As $\forall i \in \mathcal{R}_t = \mathcal{F} - \mathcal{A}_t, |x_i^T\theta_t^*| < 1, \theta_t^* \in \Omega_{\mathcal{F}}$. With $\theta^* = \sup_{\theta\in\Omega_{\mathcal{F}}} D(\theta)$, we get $D(\theta^*) \geq D(\theta_t^*)$. As we already know $D(\theta^*) \leq D(\theta_t^*)$, we then have $D(\theta^*) = D(\theta_t^*)$. Since the dual problem is convex and smooth, and the feasible set is closed and convex, $\theta_t^* = \theta^*$. Hence, $\beta_t^* \in S_{\mathcal{F}}$ as the primal solution may not be unique.

d) For ADD operations, we choose a feature in $\mathcal{R}_t$ that is mostly correlated to the residual dual variables, that is $\max_{i\in\mathcal{R}_t} |\mathbf{x}_i^T\theta_t^*|$. With feature $i \in \mathcal{R}_t$ and $\theta_t^* \in B(\theta_t, r_t)$, we have $\left||x_i^T\theta_t| - ||x_i||_2 r_t\right| \leq |x_i^T\theta_t^*| \leq |x_i^T\theta_t| + ||x_i||_2 r_t$ by the Pythagorean theorem. Thus $\forall j \in \mathcal{R}_t, j \neq i, \left||x_i^T\theta| - ||x_i||_2 r_t\right| \geq |x_j^T\theta| + ||x_j||_2 r_t$, and $|x_i^T\theta_t^*| \geq |x_j^T\theta_t^*|, \forall j \in \mathcal{R}_t, j \neq i$.

Remark 1 *Theorem 1-c) provides us the stopping criterion for ADD operations in our SAIF algorithm. We can apply ADD and DEL operations in iterations to minimize* $\max_{i\in\mathcal{R}_t} |x_i^T\theta_t^*|$ *until* $\max_{i\in\mathcal{R}_t} |x_i^T\theta_t^*| < 1$*. Hence, with* $B(\theta_t, r_t) = \{\theta_t^* \big| \, ||\theta_t^* - \theta_t||_2 \leq r_t\}$*, if we have* $\max_{i\in\mathcal{R}_t} |x_i^T\theta_t| + ||x_i||_2 r_t < 1$*, we can stop ADD operations.*

Remark 2 *Moreover, if* $\forall j \in \mathcal{R}_t$*,* $|x_j^T\theta_t^*| < 1$*, from Theorem 1-c), we can see that* $\theta_t^* = \theta^*$*, thus* $\bar{\mathcal{A}} \subseteq \mathcal{A}_t$*. So if* $\bar{\mathcal{A}} \nsubseteq \mathcal{A}_t$*,* $\exists j \in \mathcal{R}_t, |x_j^T\theta_t^*| \geq 1$*. This concludes that our*

stopping criterion for ADD operations ensures safe feature screening.

The DEL operation is similar to the screening steps in dynamic screening. As we can see, at step t with the DEL operation, $D(\theta_t^*) = D(\theta_{t-1}^*)$. Theorem 1-a) implies that $D(\theta_t^*) \leq D(\theta_{t-1}^*)$. Thus the optimal dual objective value always goes down. Theorem 1-c) and Remark 1 show that after the stopping of ADD operation, $\mathcal{A}_t$ already have recruited all of the active features for the original problem. After this the algorithm stops once it reaches the pre-specified accuracy value of the duality gap. Such monotonicity leads to the convergence of SAIF detailed in Section 3.

2.2.2 Implementation

We first discuss how we derive a tighter ball region $B(\theta_t, r_t)$ for the rage estimate of θ_t^*, taking advantages of existing screening methods.

Dual variable range estimation: Accurately estimating the range of θ_t^*, $B(\theta_t, r_t)$, for the sub-problem is critical for efficient SAIF screening with ADD and DEL operations at each iteration. With $\mathbf{f}$ as the vector form of loss function regarding all of the samples, we provide the following theorem to estimate the ball region for θ_t^* with the similar idea from sequential screening.

Theorem 2 *For the LASSO problem with the loss function* $\mathbf{f}$*, if* $\mathbf{f}^*$ *is* $\frac{1}{\alpha}$*-strongly convex, and* θ_0^* *and* θ^* *are the optimal solutions to the dual problem at* λ_0 *and* λ *with* $\lambda < \lambda_0$*, then*

$$||\theta^* - \frac{\lambda_0}{\lambda}\theta_0^*||_2^2 \leq \frac{2\alpha}{\lambda^2}\left[\mathbf{f}^*(-\frac{\lambda^2}{\lambda_0}\theta_0^*) - \mathbf{f}^*(-\lambda_0\theta_0^*) + (\lambda - \lambda_0)\langle \mathbf{f}'^*(-\lambda_0\theta_0^*), \theta_0^*\rangle\right]. \quad (2.3)$$

If we have $\theta \in \Omega$*, the bound can be further improved by*

$$||\theta^* - \frac{\lambda_0}{\lambda}\theta_0^*||_2^2 \leq \frac{2\alpha}{\lambda^2}\left[\mathbf{f}^*(-\lambda\bar{\theta}(\bar{\varrho})) - \mathbf{f}^*(-\lambda_0\theta_0^*) + (\lambda - \lambda_0)\langle \mathbf{f}'^*(-\lambda_0\theta_0^*), \theta_0^*\rangle\right], \quad (2.4)$$

where $\bar{\theta}(\bar{\varrho}) = (1 - \bar{\varrho})\theta + \bar{\varrho}\frac{\lambda}{\lambda_0}\theta_0^*$*, and* $\bar{\varrho} = argmin_{\varrho:0\leq\varrho\leq1}\mathbf{f}^*(-\lambda\bar{\theta}(\varrho))$.

Proof: As f^* is $\frac{1}{\alpha}$-strongly convex, we have

$$||\lambda\theta^* - \lambda_0\theta_0^*||_2^2 \leq 2\alpha\Big[\mathbf{f}^*(-\lambda\theta^*) - \mathbf{f}^*(-\lambda_0\theta_0^*) - \langle\mathbf{f}'^*(-\lambda_0\theta_0^*), -\lambda\theta^* - (-\lambda_0\theta_0^*)\rangle\Big],$$

which is

$$||\theta^* - \frac{\lambda_0}{\lambda}\theta_0^*||_2^2 \leq \frac{2\alpha}{\lambda^2}\Big[\mathbf{f}^*(-\lambda\theta^*) - \mathbf{f}^*(-\lambda_0\theta_0^*) + \langle\mathbf{f}'^*(-\lambda_0\theta_0^*), \lambda\theta^* - \lambda_0\theta_0^*\rangle\Big]. \tag{2.5}$$

As θ_0^* is the optimal solution at λ_0 we can see $\theta_0^* \in \Omega$, and $\frac{\lambda}{\lambda_0}\theta_0^* \in \Omega$, thus

$$\mathbf{f}^*(-\lambda\theta^*) \leq \mathbf{f}^*(-\lambda\frac{\lambda}{\lambda_0}\theta_0^*) \tag{2.6}$$

Also as θ_0^* is the optimal dual solution at λ_0, thus we have

$$\langle-\lambda_0\mathbf{f}'^*(-\lambda_0\theta_0^*), \theta^* - \theta_0^*\rangle \geq 0 \tag{2.7}$$

$$\Longrightarrow \langle-\mathbf{f}'^*(-\lambda_0\theta_0^*), \lambda\theta^* - \lambda\theta_0^*\rangle \geq 0 \tag{2.8}$$

$$\Longrightarrow \langle\mathbf{f}'^*(-\lambda_0\theta_0^*), \lambda\theta^*\rangle \leq \langle\mathbf{f}'^*(-\lambda_0\theta_0^*), \lambda\theta_0^*\rangle \tag{2.9}$$

With (2.5)- (2.9), we have

$$||\theta^* - \frac{\lambda_0}{\lambda}\theta_0^*||_2^2 \leq \frac{2\alpha}{\lambda^2}\Big[\mathbf{f}^*(-\frac{\lambda^2}{\lambda_0}\theta_0^*) - \mathbf{f}^*(-\lambda_0\theta_0^*) + (\lambda - \lambda_0)\langle\mathbf{f}'^*(-\lambda_0\theta_0^*), \theta_0^*\rangle\Big]. \tag{2.10}$$

As $\theta \in \Omega$, we have $\bar{\theta} = (1 - \varrho)\theta + \varrho\frac{\lambda}{\lambda_0}\theta_0^* \in \Omega, if\ 0 \leq \varrho \leq 1$, which implies

$$\mathbf{f}^*(-\lambda\theta^*) \leq \min_{\varrho:0\leq\varrho\leq1} \mathbf{f}^*(-\lambda\bar{\theta}(\varrho)) \leq \mathbf{f}^*(-\frac{\lambda^2}{\lambda_0}\theta_0^*). \tag{2.11}$$

So we have improved the bound as

$$||\theta^* - \frac{\lambda_0}{\lambda}\theta_0^*||_2^2 \leq \frac{2\alpha}{\lambda^2}\Big[\mathbf{f}^*(-\lambda\bar{\theta}(\bar{\varrho})) - \mathbf{f}^*(-\lambda_0\theta_0^*) + (\lambda - \lambda_0)\langle\mathbf{f}'^*(-\lambda_0\theta_0^*), \theta_0^*\rangle\Big], \quad (2.12)$$

where $\bar{\theta}(\bar{\varrho}) = (1-\bar{\varrho})\theta + \bar{\varrho}\frac{\lambda}{\lambda_0}\theta_0^*$, and $\bar{\varrho} = argmin_{\varrho:0\leq\varrho\leq 1}\mathbf{f}^*(-\lambda\bar{\theta}(\varrho))$.

At step t with the active set $\mathcal{A}_t$, $\lambda_{max(t)}$ is the minimum λ that leads to $\beta_t^* = 0$. It is easy to compute $\lambda_{max(t)} = \max_{i\in\mathcal{A}_t}|x_i^T\mathbf{f}'(0)|$, and $\theta_{0(t)}^* = -\frac{\mathbf{f}'(0)}{\lambda_{0(t)}}$. If we take $\lambda_{0(t)} = \lambda_{max(t)}$, we can use Theorem 2 to estimate θ_t^*. For linear regression, the estimation can be further improved based on the projection properties as in DPP [11].

Theorem 2 provides a tight estimation when λ_0 is close to λ. When λ is far away from λ_0, we can adopt the tighter dual variable range estimation with the following ball region by dynamic screening [38, 39]. At step t, we have

$$\forall\theta_t \in \Omega_{\mathcal{A}_t}, \beta_t \in R^{p_t\times 1}, \;\; ||\theta_t^* - \theta_t||_2^2 \leq \frac{2}{\lambda^2}\Big[P_t(\beta_t) - D(\theta_t)\Big]. \quad (2.13)$$

For β_t, with the primal-dual relation, we can easily project it to the dual feasible region $\Omega_{\mathcal{A}_t}$ to get a feasible dual variable θ_t.

With two ball regions from Theorem 2 and the duality gap, we can derive a tighter constrained region by computing the center and radius of a ball region $B(\theta_t, r_t)$ that covers the intersection of two ball regions, $B_1(\theta_1, r_1)$ and $B_2(\theta_2, r_2)$:

$$\begin{aligned} &r_t = \frac{2A}{d}, \quad \theta_t = (1-\frac{d_1}{d})\theta_1 + \frac{d_1}{d}\theta_2, \quad d_1 = \sqrt{r_1^2 - r_t^2} \\ &d = ||\theta_1 - \theta_2||_2, \quad A = \sqrt{s(s-r_1)(s-r_2)(s-d)}, \quad s = \frac{r_1 + r_2 + d}{2}, \end{aligned} \quad (2.14)$$

where B_1 can be derived from Theorem 2, and B_2 from (2.13). The resulting $B(\theta_t, r_t)$ gives us a tighter region at step t when $r_t < \min\{r1, r2\}$. When we do not have the

solutions with other λ values, we simply set the bounding region for θ_t^* based on (2.13).

Improve SAIF with a factor of the estimation: The estimation of dual variables may be inaccurate to have enough screening power during the optimization iterations, especially at the beginning of the algorithm. We add a factor to the radius of the ball region to reduce redundant computation resulted from inaccurately recruited features. At the beginning of Algorithm 1, δ is a value smaller than 1. δ will be increased to 1 during the SAIF iterations to ensure the safe guarantee of SAIF algorithm.

ADD operation implementation details: The number of added features in each ADD operation can vary to reduce redundant iterations. Generally, the relationship between the screening power and this number depends on the regularization parameter λ and how well feature vectors $x_i,\ i \in \mathcal{F}$, correlate with the outcome label y. In this chapter, we empirically set the number to be $h = \lceil c \log(\frac{md+mx}{\lambda}) \log(p) \rceil$. Here mx and md are the maximum and median of $|X^T \mathbf{f}'(\mathbf{0})|$ ($|X^T \mathbf{y}|$ with linear regression). Many iterations may need to be operated to reach the dual space point that can distinguish h features, and this may reduce the efficiency of the algorithm. We can decrease the redundancy by relaxing the strict condition in Theorem 1-d). Let S_j represent the set of features that violate the condition in Theorem 1-d) regarding feature j, i.e., $S_j = \{k \big| k \in \mathcal{R}_t, k \neq j, \big| |x_j^T \theta_t| - ||x_j||_2 r_t \big| \leq |x_k^T \theta_t| + ||x_k||_2 r_t\}$. For a feature $j \in R_t$, if $|S_j| < \tilde{h}$, we move it from R_t to A_t. Here $\tilde{h} = \lceil \zeta h \rceil$, and $\zeta > 0$. Algorithm 6 summarizes the implementation of the ADD

operation.

Data: θ_t, r_t, R_t, A_t, X

Result: R_{t+1}, A_{t+1}

Set $h = \lceil c \log(\frac{md+mx}{\lambda}) \log(p) \rceil$;

$\tilde{h} = \lceil \zeta h \rceil$;

for $i = 1$ **to** h **do**

  $j \leftarrow \max_{l \in R_t} |x_l^T \theta_t|$;

  Set $S_j = \{k \big| k \in \mathcal{R}_t, k \neq j, |x_k^T \theta| + ||x_k||_2 r \geq \big| |x_j^T \theta| - ||x_j||_2 r \big| \}$;

  if $|S_j| < \tilde{h}$ **then**

    $A_t \leftarrow A_t \cup \{j\}$;

    $R_t \leftarrow R_t - \{j\}$;

  else

    Stop;

  end

end

$A_{t+1} \leftarrow A_t$;

$R_{t+1} \leftarrow R_t$;

Algorithm 2: Algorithm for ADD operation

2.3 Convergence Analysis

In this section, we first discuss the convergence properties of SAIF and then provide the detailed complexity analysis of our SAIF algorithm.

2.3.1 Algorithm Properties

Similar to dynamic screening, SAIF employs coordinate minimization (CM) in the primal variable space. Besides feature screening (DEL), SAIF has feature recruiting operation (ADD). In this subsection, we first discuss the convergence of the base algorithm,

then we show that the number DEL and ADD operations are finite in SAIF.

2.3.1.1 Coordinate Minimization (CM)

The base algorithm we employ in SAIF is shooting algorithm [18], which is a cyclic block coordinate minimization method. Coordinate descent (CD) and coordinate minimization (CM) methods have been studied by many researchers [50, 51, 52]. Recently [53] gives faster convergence estimations for coordinate descent and CM methods on convex problems. Based on the analysis from [53], we can prove the following lemma regarding CM for LASSO. We use k to indicate the iteration or base operation number of CM, and t for the iteration number in the outer loop of SAIF or dynamic screening.

Lemma 1 (Adaptation of [53]) *For the LASSO problem with a γ-convex loss function, with cyclic coordinate minimization at most $\log_{\psi} \frac{\varepsilon}{P(\beta_0)-P(\beta^*)}$ base operations are performed to arrive at β_a that $P(\beta_a) - P(\beta^*) \leq \varepsilon$, where $\psi = \frac{\gamma^2}{p\bar{L}^2+\gamma^2}$, $\bar{L} = \sqrt{\sigma_{\max}}L$, $\sigma_{\max}$ is largest eigenvalue of X^TX, L is the Lipschitz constant of $\mathbf{f}'$, and β_0 is the starting point.*

Proof: With L as the Lipschitz constant of $\mathbf{f}'$, then $\bar{L} = \sqrt{\sigma_{\max}}L$ is the Lipschitz constant of $X^T\mathbf{f}'$. Following the proof of Theorem 8 by [53], we have

$$P(\beta_{k+1}) - P(\beta^*) \leq \frac{p\bar{L}^2}{2\gamma}||\beta_{k+1} - \beta_k||_2^2 \,. \tag{2.15}$$

Then

$$P(\beta_k) - P(\beta^*) = P(\beta_k) - P(\beta_{k+1}) + P(\beta_{k+1}) - P(\beta^*) \tag{2.16}$$

$$\geq \frac{\gamma}{2}||\beta_k - \beta_{k+1}||_2^2 + P(\beta_{k+1}) - P(\beta^*) \tag{2.17}$$

$$\geq (1 + \frac{\gamma^2}{p\bar{L}^2})\big(P(\beta_{k+1}) - P(\beta^*)\big) \tag{2.18}$$

Thus

$$\frac{P(\beta_{k+1}) - P(\beta^*)}{P(\beta_k) - P(\beta^*)} \leq \frac{p\bar{L}^2}{p\bar{L}^2 + \gamma^2} = \psi \ . \tag{2.19}$$

Recursively apply (2.19), we have

$$\frac{P(\beta_a) - P(\beta^*)}{P(\beta_0) - P(\beta^*)} \leq \psi^a = \frac{\varepsilon}{P(\beta_0) - P(\beta^*)} \tag{2.20}$$

$$(P(\beta_0) - P(\beta^*))\psi^a = \varepsilon \tag{2.21}$$

And this leads to $a = \log_\psi \frac{\varepsilon}{P(\beta_0)-P(\beta^*)}$. For any iteration number $k \geq a$, we always have the primal gap $P(\beta_k) - P(\beta^*) \leq \varepsilon$.

The base operation (soft-thresholding) in CM is operated in the primal variable space. Feature screening or feature selection operations such as ADD and DEL operations are relying on the dual variable estimation. We provide the following lemma to show that the accuracy of dual variables are almost linearly bounded by the the accuracy of primal variables when the iteration number is large.

Lemma 2 *For the primal problem and dual problem, let* $\hat{\theta}_k = -\frac{\mathbf{f}'(X\beta_k)}{\lambda}$, $\tau_k = \frac{1}{\max_i |x_i^T \hat{\theta}_k|}$, *and* $\theta_k = \tau_k \hat{\theta}_k$, *with a large* k *in coordinate minimization, we have* $||\theta_k - \theta^*||_2^2 \leq \frac{1+v}{\lambda^2}||\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)||_2^2 \leq \frac{L(1+v)}{\lambda^2}||\beta_k - \beta^*||_\Sigma^2$, *where* $\Sigma = X^T X$, *and* v *is a small positive value.*

Proof: Let $\tau_k = \frac{1}{\max_i |x_i^T \hat{\theta}_k|} = \frac{1}{|x_m^T \hat{\theta}_k|}$, and $\hat{\theta}_k = \theta^* + \rho_k$. We have $\tau_k = \frac{1}{|x_m^T \theta^* + x_m^T \rho_k|} = \frac{1}{|x_m^T \theta^*| \pm |x_m^T \rho_k|}$. Here $\pm$ means plus or minus. With

$$\lim_{k\to\infty} ||\rho_k||_2 = \lim_{k\to\infty} ||\hat{\theta}_k - \theta^*||_2 = \lim_{k\to\infty} \frac{1}{\lambda}||\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)||_2 \tag{2.22}$$

$$\leq \lim_{k\to\infty} \frac{L}{\lambda}||\beta_k - \beta^*||_\Sigma \to 0, \tag{2.23}$$

and $\forall i \in \bar{\mathcal{A}},\ |x_i^T\theta^*| = 1$, and $\forall i \in \mathcal{F} - \bar{\mathcal{A}},\ |x_i^T\theta^*| < 1$, we always can reach a k that $|x_m^T\rho_k| < 1$. After setting $\varphi_k = \pm|x_m^T\rho_k|$, we have $\tau_k = \frac{1}{1+\varphi_k}$.

$$||\theta_k - \theta^*||_2^2 \tag{2.24}$$

$$=||\frac{\tau_k \mathbf{f}'(X\beta_k)}{\lambda} - \frac{\mathbf{f}'(X\beta^*)}{\lambda}||_2^2 \tag{2.25}$$

$$=\frac{1}{\lambda^2}||\tau_k \mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)||_2^2 \tag{2.26}$$

$$=\frac{1}{\lambda^2}||\frac{\mathbf{f}'(X\beta_k)}{1+\varphi_k} - \mathbf{f}'(X\beta^*)||_2^2 \tag{2.27}$$

$$=\frac{1}{\lambda^2}||\mathbf{f}'(X\beta_k)(1-\Phi) - \mathbf{f}'(X\beta^*)||_2^2 \tag{2.28}$$

$$=\frac{1}{\lambda^2}\langle(\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)) - \Phi\mathbf{f}'(X\beta_k), (\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)) - \Phi\mathbf{f}'(X\beta_k)\rangle \tag{2.29}$$

$$=\frac{1}{\lambda^2}||\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)||_2^2 + \frac{1}{\lambda^2}||\Phi\mathbf{f}'(X\beta_k)||_2^2 - \frac{2}{\lambda^2}\langle(\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)), \Phi\mathbf{f}'(X\beta_k)\rangle \tag{2.30}$$

$$=\frac{1}{\lambda^2}||\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)||_2^2 + \frac{\Phi^2}{\lambda^2}||\mathbf{f}'(X\beta_k)||_2^2 - \frac{2\Phi}{\lambda^2}\langle(\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)), \mathbf{f}'(X\beta_k)\rangle \tag{2.31}$$

Here $\Phi = \sum_{i=1}^{\infty}(-1)^{i+1}\varphi_k^i$. With $\Phi = \sum_{i=1}^{\infty}(-1)^{i+1}\varphi_k^i = (-\varphi_k)\sum_{i=0}^{\infty}(-\varphi_k)^i = (-\varphi_k)\frac{1}{1+\varphi_k} = -\tau_k\varphi_k$, we get

$$\begin{aligned}
||\theta_k - \theta^*||_2^2 \leq & \frac{1}{\lambda^2}||\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)||_2^2 + \frac{\tau_k^2\varphi_k^2}{\lambda^2}||\mathbf{f}'(X\beta_k)||_2^2 + && (2.32)\\
& \frac{2\tau_k\varphi_k}{\lambda^2}\langle(\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)), \mathbf{f}'(X\beta_k)\rangle && (2.33)\\
= & \frac{1}{\lambda^2}||\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)||_2^2 + \varphi_k\Psi && (2.34)\\
\leq & \frac{L}{\lambda^2}||X(\beta_k - \beta^*)||_2 + \varphi_k\Psi = \frac{L}{\lambda^2}||\beta_k - \beta^*||_\Sigma^2 + \varphi_k\Psi && (2.35)
\end{aligned}$$

where

$$\Psi = \frac{\tau_k^2 \varphi_k}{\lambda^2} ||\mathbf{f}'(X\beta_k)||_2^2 + \frac{2\tau_k}{\lambda^2} \langle (\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)), \mathbf{f}'(X\beta_k) \rangle \tag{2.36}$$

If $X\beta_k = X\beta^*$, we have $\tilde{\theta} = \theta^*$, $\theta = \theta^*$, $\mathbf{f}'(X\beta_k) = \mathbf{f}'(X\beta^*)$, and $\Psi = \frac{\tau_k^2 \varphi_k}{\lambda^2}$ $||\mathbf{f}'(X\beta_k)||_2^2$. Thus $||\theta_k - \theta^*||_2^2 \leq \frac{1+v}{\lambda^2} ||\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)||_2^2 \leq \frac{L(1+v)}{\lambda^2} ||\beta_k - \beta^*||_\Sigma^2$.

If $X\beta_k \neq X\beta^*$, for any $v > 0$, we alway can reach a k, to make $\varphi_k \Psi \leq \frac{v}{\lambda^2} ||\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)||_2^2 \leq \frac{Lv}{\lambda^2} ||\beta_k - \beta^*||_\Sigma^2$. In summary, we have

$$||\theta_k - \theta^*||_2^2 \leq \frac{1+v}{\lambda^2} ||\mathbf{f}'(X\beta_k) - \mathbf{f}'(X\beta^*)||_2^2 \leq \frac{L(1+v)}{\lambda^2} ||\beta_k - \beta^*||_\Sigma^2. \tag{2.37}$$

With Lemma 2, we can see that the estimation of dual variables relies on the accuracy of primal variables. In SAIF, the starting point for each β_t is already with relatively high accuracy as empirically there are only one or a few features different between steps t and $t-1$.

2.3.1.2 Finite number of ADD and DEL Operations

With CM as the inner base algorithm, we prove that the outer loop can stop in a finite number of steps. The ADD operation recruits more features into the active set, and thus results in decreasing optimal objective value as shown in Theorem 1. Since the DEL operation does not change the optimal objective value, the corresponding optimal dual objective function value of the sub-problem decreases monotonically and finally converges to the value of the original full-scale problem. Experimentally, for a given λ, the running time of SAIF is proportional to the size of the optimal active set $\bar{A}$. The following theorem provides the guarantee for the convergence of SAIF.

Theorem 3 *Let β_t^* and θ_t^* be the optimal primal and dual solutions for the sub-problem with the active feature set $\mathcal{A}_t$.*

a) If $\bar{\mathcal{A}} \nsubseteq \mathcal{A}_t$, and $t < t'$, then $\mathcal{A}_t \neq \mathcal{A}_{t'}$.

b) $\lim_{t\to\infty} \theta_t^* = \theta^*$; $\lim_{t\to\infty} \beta_t^* \in S_{\mathcal{F}}^*$.

c) $\exists T,\ \forall t \geq T, \theta_t^* = \theta^*$, *and* $\beta_t^* \in S_F^*$.

Proof: a) If $\bar{\mathcal{A}} \nsubseteq \mathcal{A}_t$, from Remark 2, we can see that, $\exists j \in \mathcal{R}_t, |x_j^T \theta_t^*| \geq 1$. If $\max_{j\in R_t}, |x_j^T \theta_t^*|$

> 1, we can apply the ADD operation at step t to add the most active feature to $\mathcal{A}_{t+1}$. We will have $D(\theta_t^*) > D(\theta_{t+1}^*)$. As $t < t'$, $D(\theta_t^*) > D(\theta_{t+1}^*) \geq D(\theta_{t'}^*)$, and $\mathcal{A}_t \neq A_{t'}$. If $\max_{j\in R_t}, |x_j^T \theta_t^*| = 1$, the optimal dual variable is already on the hyperplanes $|x_j^T \theta_t^*|$ = 1. From the algorithm, we can see that, with an ADD operation to move all $x_j : |x_j^T \theta_t^*| = 1$ to $\mathcal{A}_t$, the optimal dual solution will remain the same, i.e., $\theta_{t+1}^* = \theta_t^*$. The ADD operation will stop at step t, as $\max_{j\in R_{t+1}}, |x_j^T \theta_{t+1}^*| < 1$. DEL does not remove $x_j : |x_j^T \theta_t^*| = 1$ from the active set $\mathcal{A}_{t'}, \forall t' > t$, as the optimal dual variable will remain the same, and the algorithm will stop. Thus $\mathcal{A}_t \neq \mathcal{A}_{t'}$. In summary, we have $\mathcal{A}_t \neq \mathcal{A}_{t'}, \forall t' > t$.

b) At step t, if the operation is DEL, we have $P(\beta_t^*) = P(\beta_{t+1}^*)$, and $D(\theta_t^*) = D(\theta_{t+1}^*)$, as removing inactive features does not change primal and dual problems. If the operation is ADD, and $\max_{i\in\mathcal{R}_t} |x_i^T \theta_t^*| > 1$, we have $P(\beta_t^*) > P(\beta_{t+1}^*)$, and $D(\theta_t^*) > D(\theta_{t+1}^*)$. Thus $\exists m > 0, D(\theta_t^*) > D(\theta_{t+m}^*)$ for each step t, which means $D(\theta_t^*)$ will converge to a fixed value as $t \to \infty$. From a), $\mathcal{A}_t$ changes monotonously with finite combinations. Thus SAIF will stop within finite steps. Let $\lim_{t\to\infty} D(\theta_t^*) = \bar{d}$, and let $\Gamma = \{\theta | D(\theta) = \bar{d}, \theta \in \lim_{t\to\infty} \Omega_{\mathcal{A}_t}\}$. As $\Omega_{\mathcal{A}_t} \supseteq \Omega_{\mathcal{F}}$, we have $\bar{d} \geq D(\theta^*)$. If $\theta^* \notin \Gamma$, as the dual objective function is smooth and convex, and $\Omega_{\mathcal{F}} \subseteq \lim_{t\to\infty} \Omega_{\mathcal{A}_t}$, $\forall \hat{\theta}^* \in \Gamma, D(\hat{\theta}^*) = \bar{d} > D(\theta^*)$. As $\theta^* = argmax_{\theta\in\Omega_{\mathcal{F}}} D(\theta)$, and θ^* is unique, we have $\forall \hat{\theta}^* \in \Gamma, \hat{\theta}^* \notin \Omega_{\mathcal{F}}$. This implies $\forall \hat{\theta}^* \in \Gamma, \exists j, |x_j^T \hat{\theta}^*| > 1$, which contradicts the algorithm stopping criterion. Therefore we have $\theta^* \in \Gamma$. As the optimal dual value is unique, $\lim_{t\to\infty} \theta_t^* = \theta^*$ and $\lim_{t\to\infty} \beta_t^* \in S_{\mathcal{F}}^*$.

c) As $\Omega_{\mathcal{A}_t} = \cap_{i\in\mathcal{A}_t} \{\theta : |x_i^T \theta| \leq 1\}$, the active sets at different iterations are different before the algorithm stops from a). From b), we have $\lim_{t\to\infty} \theta_t^* = \theta^*$. There are at most

$\left(\sum_{k=0}^{k=n_A-1}\binom{n_{\mathcal{A}}}{k}\right)\left(\sum_{k=0}^{k=n_{\mathcal{R}}}\binom{n_{\mathcal{R}}}{k}\right)$ different potential active sets ($n_{\mathcal{A}}+n_{\mathcal{R}}=p, n_{\mathcal{A}}=|\bar{\mathcal{A}}|$) through the algorithm iterations of upating the current active features. In practice, the number of legitimate active set combinations is much smaller. Thus, $\exists T,\ \forall t \geq T, \theta_t^* = \theta^*$, and $\beta_t^* \in S_F^*$.

2.3.2 Complexity Analysis

For complexity analysis, we split the SAIF algorithm into three phases: feature recruiting, inactive feature deletion, and accuracy pursuing. The inactive feature deletion phase is the same as the feature screening phase in dynamic screening. We first present the complexity analysis for dynamic screening, which is our additional contribution in this manuscript, and then based on that and previous results, we give the detailed complexity analysis for SAIF.

2.3.2.1 Complexity Analysis for Dynamic Screening

Dynamic screening [38, 39] starts its active set with the whole feature set. Let r_i be the radius of the ball region for the screening of feature i, according to DEL operation,

$$|x_i^T\theta_t| + ||x_i||_2 r_i < 1 \implies r_i < \frac{1-\frac{|x_i^T\hat{\theta}_t|}{\max_j |x_j^T\hat{\theta}_t|}}{||x_i||_2} = \frac{1-\frac{|x_i^T\hat{\theta}_t|}{|x_m^T\hat{\theta}_t|}}{||x_i||_2}. \tag{2.38}$$

Here x_m is the feature with the value of $\max_j |x_j^T\hat{\theta}_t|$, $\hat{\theta}_t = -\frac{\mathbf{f}'(X\beta_t)}{\lambda}$, $\theta_t = \tau\hat{\theta}_t$, and $\tau = \frac{1}{\max_j |x_j^T\hat{\theta}_t|}$. If feature i does not belong to the final active set $\bar{\mathcal{A}}$, then $|x_i\theta^*| < 1$. With large t, x_m belongs to $\bar{\mathcal{A}}$ according to Theorem 1, and $|x_m\theta^*| = 1$. We have

$$r_i < \frac{1-\frac{|x_i^T\hat{\theta}_t|}{|x_m^T\hat{\theta}_t|}}{||x_i||_2} \approx \frac{1-|x_i^T\theta^*|}{||x_i||_2}. \tag{2.39}$$

Thus the screening radius for feature i is determined by how close $\hat{\theta}_t$ and θ^* are, linearly determined by the primal variable accuracy according to Lemma 2. With ε as the pre-specified objective function value accuracy, the following theorem gives the time complexity of the dynamic screening procedure.

Theorem 4 *Assume that the time complexity for one operation of coordinate minimization is $O(u)$, then the time complexity for dynamic screening is $O\Big(u\frac{\bar{L}^2}{\gamma^2}\big(p\log\frac{G_0}{\varepsilon_D}+|\bar{\mathcal{A}}|\log\frac{\varepsilon_D}{\varepsilon}\big)\Big)$. Here $G_0 = P(\beta_0) - P(\beta^*)$, and ε_D is the accuracy of the objective function value for the last feature screening operation.*

Proof: The computation of dynamic screening has two main phases, feature screening and accuracy pursuing, denoted by T_a and T_b respectively. Let $G_t = P(\beta_t) - P(\beta^*)$ to represent the primal accuracy after t outer loop iterations. Ku is the complexity for K CM iterations, and we need np_t to compute the duality gap. Let Z to represent the total number of outer loop iterations for the feature screening phase. Then we have

$$T_a = \sum_{t=1}^{Z} \frac{\log_{\psi_t}\frac{G_t}{G_{t-1}}}{K}(Ku + np_t), \text{and } T_b = u\log_{\psi_Z}\frac{\varepsilon}{G_Z}. \tag{2.40}$$

The complexity is

$$\begin{aligned} T &= T_a + T_b && (2.41)\\ &= \sum_{t=1}^{Z} \frac{\log_{\psi_t}\frac{G_t}{G_{t-1}}}{K}(Ku + np_t) + u\log_{\psi_Z}\frac{\varepsilon}{G_Z} && (2.42)\\ &= u\sum_{t=1}^{Z}\log_{\psi_t}\frac{G_t}{G_{t-1}} + \frac{n}{K}\sum_{t=1}^{Z}\log_{\psi_t}\frac{G_t}{G_{t-1}}p_t + u\log_{\psi_Z}\frac{\varepsilon}{G_Z} && (2.43)\\ &= u\sum_{t=1}^{Z}\log_{\psi_t}\frac{G_t}{G_{t-1}} + u\log_{\psi_Z}\frac{\varepsilon}{G_Z} + \frac{n}{K}\sum_{t=1}^{Z}\log_{\psi_t}\frac{G_t}{G_{t-1}}p_t. && (2.44) \end{aligned}$$

By following the proof of Theorem 3 in [53],

$$\log_{\psi_t} \frac{G_t}{G_{t-1}} \leq \frac{p_t \bar{L}^2 + \gamma^2}{\gamma^2} (\log \frac{G_{t-1}}{G_t}), \tag{2.45}$$

we have

$$T_1 = u \sum_{t=1}^{Z} \log_{\psi_t} \frac{G_t}{G_{t-1}} + u \log_{\psi_Z} \frac{\varepsilon}{G_Z} \tag{2.46}$$

$$\leq u \sum_{t=1}^{Z} (1 + \frac{p_t \bar{L}^2}{\gamma^2}) \log \frac{G_{t-1}}{G_t} + u(1 + \frac{|\bar{\mathcal{A}}| \bar{L}^2}{\gamma^2}) \log \frac{G_z}{\varepsilon} \tag{2.47}$$

$$= u \log \frac{G_0}{\varepsilon} + \frac{u\bar{L}^2}{\gamma^2} \log \big(\Pi_{t=1}^{Z} \frac{G_{t-1}^{p_t}}{G_t^{p_t}} \big) \frac{G_z^{|\bar{\mathcal{A}}|}}{\varepsilon^{|\bar{\mathcal{A}}|}} = u \log \frac{G_0}{\varepsilon} + \frac{u\bar{L}^2}{\gamma^2} \log \frac{G_0^p}{\bar{G}^{p-|\bar{\mathcal{A}}|} \varepsilon^{|\bar{\mathcal{A}}|}} \tag{2.48}$$

$$= u \log \frac{G_0}{\varepsilon} + \frac{u\bar{L}^2}{\gamma^2} \Big((p - |\bar{\mathcal{A}}|) \log \frac{G_0}{\bar{G}} + |\bar{\mathcal{A}}| \log \frac{G_0}{\varepsilon} \Big). \tag{2.49}$$

Here $\bar{G} = \big(\Pi_{t=1}^{Z-1} G_t^{p_t - p_{t+1}} G_Z^{p_{Z-1} - |\bar{\mathcal{A}}|} \big)^{\frac{1}{p - |\bar{\mathcal{A}}|}}$.

$$T_2 = \frac{n}{K} \sum_{t=1}^{Z} \log_{\psi_t} \frac{G_t}{G_{t-1}} p_t \leq \frac{n}{K} \sum_{t=1}^{Z} (p_t + \frac{p_t^2 \bar{L}^2}{\gamma^2}) \log \frac{G_{t-1}}{G_t} \tag{2.50}$$

$$= \frac{n}{K} \Big(\log \frac{G_0^p}{\bar{G}^{p-|\bar{\mathcal{A}}|} G_Z^{|\bar{\mathcal{A}}|}} + \frac{\bar{L}^2}{\gamma^2} \log \frac{G_0^{p^2}}{\tilde{G}^{p^2 - |\bar{\mathcal{A}}|^2} G_Z^{|\bar{\mathcal{A}}|^2}} \Big), \tag{2.51}$$

where $\tilde{G} = \big(\Pi_{t=1}^{Z-1} G_t^{p_t^2 - p_{t+1}^2} G_Z^{p_{Z-1}^2 - |\bar{\mathcal{A}}|^2} \big)^{\frac{1}{p^2 - |\bar{\mathcal{A}}|^2}}$.

As

$$\bar{G} \geq \big(\Pi_{t=1}^{Z-1} G_Z^{p_t - p_{t+1}} G_Z^{p_{Z-1} - |\bar{\mathcal{A}}|} \big)^{\frac{1}{p - |\bar{\mathcal{A}}|}} = G_Z, \tag{2.52}$$

and

$$\tilde{G} \geq \left(\Pi_{t=1}^{Z-1} G_Z^{p_t^2 - p_{t+1}^2} G_Z^{p_{Z-1}^2 - |\bar{\mathcal{A}}|^2}\right)^{\frac{1}{p^2 - |\bar{\mathcal{A}}|^2}} = G_Z. \tag{2.53}$$

$$\begin{aligned}
T =& T_1 + T_2 && (2.54)\\
\leq& u \log \frac{G_0}{\varepsilon} + \frac{u\bar{L}^2}{\gamma^2}\Big((p - |\bar{\mathcal{A}}|)\log\frac{G_0}{G_Z} + |\bar{\mathcal{A}}|\log\frac{G_0}{\varepsilon}\Big) + \frac{n}{K}\Big(\log\frac{G_0^p}{G_Z^{p-|\bar{\mathcal{A}}|}G_Z^{|\bar{\mathcal{A}}|}} + && (2.55)\\
& \frac{\bar{L}^2}{\gamma^2}\log\frac{G_0^{p^2}}{G_Z^{p^2-|\bar{\mathcal{A}}|^2}G_Z^{|\bar{\mathcal{A}}|^2}}\Big) && (2.56)\\
=& u \log \frac{G_0}{\varepsilon} + \frac{u\bar{L}^2}{\gamma^2}\Big((p - |\bar{\mathcal{A}}|)\log\frac{G_0}{G_Z} + |\bar{\mathcal{A}}|\log\frac{G_0}{\varepsilon}\Big) + \frac{n}{K}\Big(p\log\frac{G_0}{G_Z} + && (2.57)\\
& \frac{p^2\bar{L}^2}{\gamma^2}\log\frac{G_0}{G_Z}\Big). && (2.58)
\end{aligned}$$

We can set $K = Cp$, with $G_Z = \varepsilon_D \psi_Z^k$, $k \in \{1, 2, ..., K\}$, we have

$$\begin{aligned}
T \leq& u \log \frac{G_0}{\varepsilon} + \frac{u\bar{L}^2}{\gamma^2}\Big((p - |\bar{\mathcal{A}}|)\log\frac{G_0}{G_Z} + |\bar{\mathcal{A}}|\log\frac{G_0}{\varepsilon}\Big) + \frac{n}{K}\Big(p\log\frac{G_0}{G_Z} && (2.59)\\
&+ \frac{p^2\bar{L}^2}{\gamma^2}\log\frac{G_0}{G_Z}\Big) && (2.60)\\
=& u \log \frac{G_0}{\varepsilon} + \frac{u\bar{L}^2}{\gamma^2}\Big((p - |\bar{\mathcal{A}}|)\log\frac{G_0}{G_Z} + |\bar{\mathcal{A}}|\log\frac{G_0}{\varepsilon}\Big) + \frac{n}{C}\Big(\log\frac{G_0}{G_Z} + \frac{p\bar{L}^2}{\gamma^2}\log\frac{G_0}{G_Z}\Big) && (2.61)\\
=& \Big(u\frac{\bar{L}^2}{\gamma^2}(p - |\bar{\mathcal{A}}| + \frac{p}{C}) + \frac{n}{C} + \frac{n}{C}\frac{p\bar{L}^2}{\gamma^2}\Big)\log\frac{G_0}{G_Z} + u(1 + \frac{\bar{L}^2}{\gamma^2}|\bar{\mathcal{A}}|)\log\frac{G_0}{\varepsilon} && (2.62)\\
=& \Big(u\frac{\bar{L}^2}{\gamma^2}(p + \frac{p}{C}) + u + \frac{n}{C} + \frac{n}{C}\frac{p\bar{L}^2}{\gamma^2}\Big)\log\frac{G_0}{G_Z} + u(1 + \frac{\bar{L}^2}{\gamma^2}|\bar{\mathcal{A}}|)\log\frac{G_Z}{\varepsilon} && (2.63)\\
=& up\eta\frac{\bar{L}^2}{\gamma^2}\log\frac{G_0}{G_Z} + u\frac{\bar{L}^2}{\gamma^2}|\bar{\mathcal{A}}|\log\frac{G_Z}{\varepsilon} + u\log\frac{G_Z}{\varepsilon} + (u + \frac{n}{C})\log\frac{G_0}{G_Z}. && (2.64)
\end{aligned}$$

Here $\eta = 1 + \frac{1}{C} + \frac{u}{Cn}$. With $\epsilon_D = G_Z$, ignoring the last two terms, the complexity of dynamic screening can be simplified as $O\Big(u\frac{\bar{L}^2}{\gamma^2}\big(p\log\frac{G_0}{\varepsilon_D} + |\bar{\mathcal{A}}|\log\frac{\varepsilon_D}{\varepsilon}\big)\Big)$.

Remark 3 *With coordinate minimization, the number of iterations to reach the accuracy of the objective function value ϵ is $O\Big(\frac{\bar{L}^2}{\gamma^2}\big(p\log\frac{1}{\varepsilon_D} + |\bar{\mathcal{A}}|\log\frac{\varepsilon_D}{\varepsilon}\big)\Big)$. As $p >> |\bar{\mathcal{A}}|$, the computation cost in dynamic screening is mainly from the iterations to reach ϵ_D.*

Experiments will confirm the conclusions from Theorem 4 and Remark 3 in the results presented in Section 5.

2.3.2.2 Complexity Analysis for SAIF

With the complexity analysis for dynamic screening, we now derive the complexity of SAIF and show its advantages over dynamic screening theoretically. SAIF starts the algorithm from the feature recruiting phase. The ADD operation recruit a feature with $\max_{i\in\mathcal{R}_t}|x_i^T\theta_t|$. When θ_t is close to θ_t^*, we have

$$|x_i^T\theta_t| - ||x_i||_2 r_i > |x_k^T\theta_t| + ||x_k||_2 r_i,\ \forall k \in \mathcal{R}_t, k \neq i \tag{2.65}$$

$$\Longrightarrow r_i < \frac{|x_i^T\theta_t| - |x_k^T\theta_t|}{||x_i||_2 + ||x_k||_2} \approx \frac{|x_i^T\theta_t^*| - |x_k^T\theta_t^*|}{||x_i||_2 + ||x_k||_2}\ \forall k \in \mathcal{R}_t, k \neq i. \tag{2.66}$$

Here we use θ_t^* rather than θ^* as the algorithm has not reached the stopping point of ADD operations and $\bar{\mathcal{A}} \nsubseteq \mathcal{A}_t$. In (2.66), the radius for adding feature i into the active set is determined by how large it can outperform the other features. We use T_a to represent the running time consumed in the feature recruiting phase. The inactive feature deletion phase starts from setting IsADD = False in SAIF in Algorithm 1. Let $Q_t(\beta) = P_t(\beta) - P_t(\beta_t^*)$, the time complexity for SAIF with CM is given by the following lemma and theorem.

Lemma 3 *With $O(u)$ as the complexity for the base operation of cyclic coordinate minimization of the LASSO problem with a γ-convex loss function, the complexity for the feature recruiting phase is*

$$T_a = \frac{Ku + pn}{K}\Big(\Upsilon + \frac{\bar{L}^2}{\gamma^2}\Phi + p_{T_I}\frac{\bar{L}^2}{\gamma^2}\log\frac{\bar{Q}}{Q_{T_I}(\beta_{T_I})}\Big), \tag{2.67}$$

where

$$\bar{Q} = \big(\Pi_{t=1}^{T_I-1} Q_{t+1}(\beta_t)^{p_{t+1}-p_t}\big)^{\frac{1}{p_{T_I}}}, \quad \Upsilon = \log\big(\Pi_{t=1}^{T_I-1} \frac{Q_{t+1}(\beta_t)}{Q_t(\beta_t)^{\frac{p_t}{p_{t+1}}}} \frac{1}{Q_{T_I}(\beta_{T_I})}\big), \tag{2.68}$$

$$\Phi = \log\big(\Pi_{t=1}^{T_I-1} \frac{Q_{t+1}(\beta_t)^{p_t}}{Q_t(\beta_t)^{p_t}}\big), \quad \textit{and ADD operation stops after } T_I \textit{ steps}. \tag{2.69}$$

Proof: The time complex for each t before stopping ADD operation is $Ku + np$.

$$T_a = \sum_{t=1}^{T_I} \frac{\log_{\psi_t} \frac{Q_t(\beta_t)}{Q_t(\beta_{t-1})}}{K}(Ku + np) \tag{2.70}$$

$$= \frac{Ku + np}{K} \sum_{t=1}^{T_I} \log_{\psi_t} \frac{Q_t(\beta_t)}{Q_t(\beta_{t-1})} \tag{2.71}$$

$$= \frac{Ku + np}{K}\Big(- \log_{\psi_1} Q_1(\beta_0) + \sum_{t=1}^{T_I-1} \big(\log_{\psi_t} Q_t(\beta_t) - \log_{\psi_{t+1}} Q_{t+1}(\beta_t)\big) \tag{2.72}$$

$$+ \log_{\psi_{T_I}} Q_{T_I}(\beta_{T_I})\Big) \tag{2.73}$$

$$= \frac{Ku + np}{K}\Big(- \log_{\psi_1} Q_1(\beta_0) + \sum_{t=1}^{T_I-1} \big(\log_{\psi_t} Q_t(\beta_t) - \log_{\psi_{t+1}} Q_{t+1}(\beta_t)\big) \tag{2.74}$$

$$+ \log_{\psi_{T_I}} Q_{T_I}(\beta_{T_I})\Big) \tag{2.75}$$

$$= \frac{Ku + np}{K}\Big(\log_{\psi_{T_I}} Q_{T_I}(\beta_{T_I}) - \log_{\psi_1} Q_1(\beta_0) + \sum_{t=1}^{T_I-1} \log_{\psi_{t+1}} \frac{Q_t(\beta_t)^{\frac{\log \psi_{t+1}}{\log \psi_t}}}{Q_{t+1}(\beta_t)}\Big) \tag{2.76}$$

$$\leq \frac{Ku + np}{K}\Big(- (1 + p_{T_I}\frac{\bar{L}^2}{\gamma^2}) \log Q_{T_I}(\beta_{T_I}) - \log_{\psi_1} Q_1(\beta_0) + \tag{2.77}$$

$$\sum_{t=1}^{T_I-1} (1 + p_{t+1}\frac{\bar{L}^2}{\gamma^2}) \log \frac{Q_{t+1}(\beta_t)}{Q_t(\beta_t)^{\frac{\log \psi_{t+1}}{\log \psi_t}}}\Big) \tag{2.78}$$

With $\frac{\log \psi_{t+1}}{\log \psi_t} \approx \frac{p_t}{p_{t+1}}$, we have

$$T_a \leq \frac{Ku+np}{K}\Big(-(1+p_{T_I}\frac{\bar{L}^2}{\gamma^2})\log Q_{T_I}(\beta_{T_I}) - \log_{\psi_1} Q_1(\beta_0) + \tag{2.79}$$

$$\sum_{t=1}^{T_I-1}(1+p_{t+1}\frac{\bar{L}^2}{\gamma^2})\log\frac{Q_{t+1}(\beta_t)}{Q_t(\beta_t)^{\frac{p_t}{p_{t+1}}}}\Big) \tag{2.80}$$

$$\leq \frac{Ku+np}{K}\Big(\log\big(\Pi_{t=1}^{T_I-1}\frac{Q_{t+1}(\beta_t)}{Q_t(\beta_t)^{\frac{p_t}{p_{t+1}}}}\frac{1}{Q_{T_I}(\beta_{T_I})}\big) \tag{2.81}$$

$$+\frac{\bar{L}^2}{\gamma^2}\log\big(\Pi_{t=1}^{T_I-1}\frac{Q_{t+1}(\beta_t)^{p_{t+1}}}{Q_t(\beta_t)^{p_t}}\big)\frac{1}{Q_{T_I}(\beta_{T_I})^{p_{T_I}}}\Big) \tag{2.82}$$

$$= \frac{Ku+np}{K}\Big(\log\big(\Pi_{t=1}^{T_I-1}\frac{Q_{t+1}(\beta_t)}{Q_t(\beta_t)^{\frac{p_t}{p_{t+1}}}}\frac{1}{Q_{T_I}(\beta_{T_I})}\big) + \frac{\bar{L}^2}{\gamma^2}\log\big(\Pi_{t=1}^{T_I-1}\frac{Q_{t+1}(\beta_t)^{p_t}}{Q_t(\beta_t)^{p_t}}\big) + \tag{2.83}$$

$$\frac{\bar{L}^2}{\gamma^2}\log\frac{\Pi_{t=1}^{T_I-1}Q_{t+1}(\beta_t)^{p_{t+1}-p_t}}{Q_{T_I}(\beta_{T_I})^{p_{T_I}}}\Big) \tag{2.84}$$

$$= \frac{Ku+np}{K}\Big(\log\big(\Pi_{t=1}^{T_I-1}\frac{Q_{t+1}(\beta_t)}{Q_t(\beta_t)^{\frac{p_t}{p_{t+1}}}}\frac{1}{Q_{T_I}(\beta_{T_I})}\big) + \frac{\bar{L}^2}{\gamma^2}\log\big(\Pi_{t=1}^{T_I-1}\frac{Q_{t+1}(\beta_t)^{p_t}}{Q_t(\beta_t)^{p_t}}\big) + \tag{2.85}$$

$$p_{T_I}\frac{\bar{L}^2}{\gamma^2}\log\frac{\bar{Q}}{Q_{T_I}(\beta_{T_I})}\Big). \tag{2.86}$$

Here

$$\bar{Q} = \big(\Pi_{t=1}^{T_I-1}Q_{t+1}(\beta_t)^{p_{t+1}-p_t}\big)^{\frac{1}{p_{T_I}}}. \tag{2.87}$$

Let

$$\Upsilon = \log\big(\Pi_{t=1}^{T_I-1}\frac{Q_{t+1}(\beta_t)}{Q_t(\beta_t)^{\frac{p_t}{p_{t+1}}}}\frac{1}{Q_{T_I}(\beta_{T_I})}\big), \text{ and } \Phi = \log\big(\Pi_{t=1}^{T_I-1}\frac{Q_{t+1}(\beta_t)^{p_t}}{Q_t(\beta_t)^{p_t}}\big). \tag{2.88}$$

This results

$$T_a = \frac{Ku + np}{K}\Big(\Upsilon + \frac{\bar{L}^2}{\gamma^2}\Phi + p_{T_I}\frac{\bar{L}^2}{\gamma^2}\log\frac{\bar{Q}}{Q_{T_I}(\beta_{T_I})}\Big). \tag{2.89}$$

Theorem 5 *With $O(u)$ as the complexity for the base operation of cyclic coordinate minimization of the LASSO problem with a γ-convex loss function, the time complexity for SAIF is $O\Big(u\frac{\bar{L}^2}{\gamma^2}\big(\bar{p}\log\frac{\bar{Q}}{\varepsilon_D} + \bar{p}p_A + |\bar{\mathcal{A}}|\log\frac{\varepsilon_D}{\varepsilon}\big)\Big)$. Here p_A is the total number of features involved in ADD operations, $\bar{p}$ is the maximum size of the active set during the algorithm iterations, $\bar{Q}$ is the geometric mean of the accuracy of the sub-problem objective function values corresponding to ADD operations, and ε_D is the accuracy of the objective function value for the last feature DEL operation.*

Proof: T_b is the time consumed by both inactive feature screening and accuracy pursuing phases. The inactive feature screening and accuracy pursue phases are similar to dynamic screening. We simplify the derivations by following the steps and techniques used in the analysis for dynamic screening.

$$T_b = \sum_{t=T_I+1}^{T_D}\frac{\log_{\psi_t}\frac{G_t}{G_{t-1}}}{K}(Ku + np_t) + u\log_{\psi_{T_D+1}}\frac{\varepsilon}{G_{T_D}} \tag{2.90}$$

$$= u\sum_{t=T_I+1}^{T_D}\log_{\psi_t}\frac{G_t}{G_{t-1}} + u\log_{\psi_{T_D+1}}\frac{\varepsilon}{G_{T_D}} + \frac{n}{K}\sum_{t=T_I+1}^{T_D}p_t\log_{\psi_t}\frac{G_t}{G_{t-1}}. \tag{2.91}$$

The first two terms can be written as

$$T_{b1} = u\sum_{t=T_I+1}^{T_D}\log_{\psi_t}\frac{G_t}{G_{t-1}} + u\log_{\psi_{T_D+1}}\frac{\varepsilon}{G_{T_D}} \tag{2.92}$$

$$\leq u\log\frac{G_{T_I}}{\varepsilon} + \frac{u\bar{L}^2}{\gamma^2}\Big((p_{T_I} - |\bar{\mathcal{A}}|)\log\frac{G_{T_I}}{\bar{G}} + |\bar{\mathcal{A}}|\log\frac{G_{T_I}}{\varepsilon}\Big). \tag{2.93}$$

Here $\bar{G} = \big(\Pi_{t=T_I+1}^{T_D-1} G_t^{p_t - p_{t+1}} G_{T_D}^{p_{T_D-1} - |\bar{\mathcal{A}}|}\big)^{\frac{1}{p_{T_I} - |\bar{\mathcal{A}}|}}$.

$$T_{b2} = \frac{n}{K} \sum_{t=T_I+1}^{T_D} p_t \log_{\psi_t} \frac{G_t}{G_{t-1}} \leq \frac{n}{K} \sum_{t=T_I+1}^{T_D} (p_t + \frac{p_t^2 \bar{L}^2}{\gamma^2}) \log \frac{G_{t-1}}{G_t} \tag{2.94}$$

$$= \frac{n}{K} \Big(\log \frac{G_{T_I}^{p_{T_I}}}{\bar{G}^{p_{T_I} - |\bar{\mathcal{A}}|} G_{T_D}^{|\bar{\mathcal{A}}|}} + \frac{\bar{L}^2}{\gamma^2} \log \frac{G_{T_I}^{p_{T_I}^2}}{\tilde{G}^{p_{T_I}^2 - |\bar{\mathcal{A}}|^2} G_{T_D}^{|\bar{\mathcal{A}}|^2}} \Big), \tag{2.95}$$

where $\tilde{G} = \big(\Pi_{t=T_I+1}^{T_D-1} G_t^{p_t^2 - p_{t+1}^2} G_{T_D}^{p_{T_D-1}^2 - |\bar{\mathcal{A}}|^2}\big)^{\frac{1}{p_{T_I}^2 - |\bar{\mathcal{A}}|^2}}$.

Similar to dynamic screening,

$$\bar{G} \geq \big(\Pi_{t=T_I+1}^{T_D-1} G_{T_D}^{p_t - p_{t+1}} G_{T_D}^{p_{T_D-1} - |\bar{\mathcal{A}}|}\big)^{\frac{1}{p_{T_I} - |\bar{\mathcal{A}}|}} = G_{T_D}, \tag{2.96}$$

and

$$\tilde{G} \geq \big(\Pi_{t=T_I+1}^{T_D-1} G_{T_D}^{p_t^2 - p_{t+1}^2} G_{T_D}^{p_{T_D-1}^2 - |\bar{\mathcal{A}}|^2}\big)^{\frac{1}{p_{T_I}^2 - |\bar{\mathcal{A}}|^2}} = G_{T_D}. \tag{2.97}$$

Thus

$$T_{b1} \leq u \log \frac{G_{T_I}}{\varepsilon} + \frac{u\bar{L}^2}{\gamma^2} \Big((p_{T_I} - |\bar{\mathcal{A}}|) \log \frac{G_{T_I}}{G_{T_D}} + |\bar{\mathcal{A}}| \log \frac{G_{T_I}}{\varepsilon} \Big), \tag{2.98}$$

and

$$T_{b2} \leq \frac{n}{K} \Big(p_{T_I} \log \frac{G_{T_I}}{G_{T_D}} + \frac{\bar{L}^2}{\gamma^2} p_{T_I}^2 \log \frac{G_{T_I}}{G_{T_D}} \Big). \tag{2.99}$$

We set K proportion to feature size for both feature increasing and inactive feature deletion phases, i.e., $K_I = Cp$ and $K_D = Cp_{T_I}$. With $G_{T_I} = Q_{T_I}(\beta_{T_I})$, the time complexity for SAIF can be written as

$$T = T_a + T_b = T_a + T_{b1} + T_{b2} \tag{2.100}$$

$$\leq \frac{K_I u + np}{K_I}\Big(\Upsilon + \frac{\bar{L}^2}{\gamma^2}\Phi + p_{T_I}\frac{\bar{L}^2}{\gamma^2}\log\frac{\bar{Q}}{G_{T_I}}\Big) + u\log\frac{G_{T_I}}{\varepsilon} + \frac{u\bar{L}^2}{\gamma^2}\Big((p_{T_I} - |\bar{\mathcal{A}}|)\log\frac{G_{T_I}}{G_{T_D}} \tag{2.101}$$

$$+ |\bar{\mathcal{A}}|\log\frac{G_{T_I}}{\varepsilon}\Big) + \frac{n}{K_D}\Big(p_{T_I}\log\frac{G_{T_I}}{G_{T_D}} + \frac{\bar{L}^2}{\gamma^2}p_{T_I}^2\log\frac{G_{T_I}}{G_{T_D}}\Big) \tag{2.102}$$

$$= (u + \frac{n}{C})\Big(\Upsilon + \frac{\bar{L}^2}{\gamma^2}\Phi + p_{T_I}\frac{\bar{L}^2}{\gamma^2}\log\frac{\bar{Q}}{G_{T_I}}\Big) + u\log\frac{G_{T_I}}{\varepsilon} + \frac{u\bar{L}^2}{\gamma^2}\Big(p_{T_I}\log\frac{G_{T_I}}{G_{T_D}} + \tag{2.103}$$

$$|\bar{\mathcal{A}}|\log\frac{G_{T_D}}{\varepsilon}\Big) + \frac{n}{C}\Big(\log\frac{G_{T_I}}{G_{T_D}} + \frac{\bar{L}^2}{\gamma^2}p_{T_I}\log\frac{G_{T_I}}{G_{T_D}}\Big) \tag{2.104}$$

$$= p_{T_I}(u + \frac{n}{C})\frac{\bar{L}^2}{\gamma^2}\log\frac{\bar{Q}}{G_{T_D}} + u\frac{\bar{L}^2}{\gamma^2}|\bar{\mathcal{A}}|\log\frac{G_{T_D}}{\varepsilon} + (u + \frac{n}{C})\frac{\bar{L}^2}{\gamma^2}\Phi \tag{2.105}$$

$$+ (u + \frac{n}{C})\Upsilon + u\log\frac{G_{T_I}}{\varepsilon} + \frac{n}{C}\log\frac{G_{T_I}}{G_{T_D}}. \tag{2.106}$$

Let $\eta = 1 + \frac{n}{uC}$, and $\mu = \max_{t:1\leq t\leq T_I-1}\eta\log\frac{P(\beta_t)-P(\beta^*_{t+1})}{P(\beta_t)-P(\beta^*_t)}$, then we have

$$\eta\Phi = \eta\log\big(\Pi_{t=1}^{T_I-1}\frac{Q_{t+1}(\beta_t)^{p_t}}{Q_t(\beta_t)^{p_t}}\big) = \sum_{t=1}^{T_I-1} p_t\eta\log\frac{Q_{t+1}(\beta_t)}{Q_t(\beta_t)} \tag{2.107}$$

$$\leq \mu\sum_{t=1}^{T_I-1} p_t \leq \mu\bar{p}p_A, \tag{2.108}$$

and

$$T \leq u\eta p_{T_I}\frac{\bar{L}^2}{\gamma^2}\log\frac{\bar{Q}}{G_{T_D}} + u\frac{\bar{L}^2}{\gamma^2}|\bar{\mathcal{A}}|\log\frac{G_{T_D}}{\varepsilon} + u\mu\bar{p}p_A + u\eta\Upsilon + u\log\frac{G_{T_I}}{\varepsilon} + \frac{n}{C}\log\frac{G_{T_I}}{G_{T_D}}. \tag{2.109}$$

Here p_A is the total number of features have been involved in the ADD operation. $\bar{p} = \max_{t:1\leq t\leq T_I} p_t$, and $\epsilon_D = G_{T_D}$, the time complexity for SAIF can be simplified as

$O\left(u\frac{\bar{L}^2}{\gamma^2}\left(\bar{p}\log\frac{\bar{Q}}{\varepsilon_D}+\bar{p}p_A+|\bar{\mathcal{A}}|\log\frac{\varepsilon_D}{\varepsilon}\right)\right)$.

Remark 4 *With coordinate minimization, the number of iterations to reach the accuracy of the objective function value ϵ is $O\left(\frac{\bar{L}^2}{\gamma^2}\left(\bar{p}\log\frac{\bar{Q}}{\varepsilon_D}+\bar{p}p_A+|\bar{\mathcal{A}}|\log\frac{\varepsilon_D}{\varepsilon}\right)\right)$. $\bar{Q}$ is a value much smaller than G_0 in dynamic screening (as the value of Q_i for adding feature i usually is very small).*

According to our experiments, $\bar{p}$ is often close to the number of the actual active features in the optimal LASSO solution, $|\bar{\mathcal{A}}|$. The dominating factor for the computational complexity of SAIF is the second term $\bar{p}p_A$. The less features being added in the active set, the less time SAIF will consume. Experimentally, p_A is often a value several times larger than $|\bar{\mathcal{A}}|$, and $p_A << p$. We can conclude that SAIF takes much less time than dynamic screening based on the analysis of Theorems 4 and 5. With the theoretical safe and convergence guarantees, SAIF can work with extremely high-dimensional data to obtain optimal LASSO solutions.

2.4 Experiments

In this section, we present the experiments comparing SAIF with other existing LASSO methods. We first evaluate the selected methods based on a simulation study and then apply them to one real-world study based on the LASSO formulation. In the second subsection, we evaluate SAIF for logistic regression with two real-world data sets. We present the comparison between SAIF and sequential screening and homotopy methods in the third subsection. The base algorithm (coordinate minimization) is implemented with C, and the main algorithm of SAIF, dynamic screening [38], DPP [11] and the homotopy method [48] is in Matlab. We use the BLITZ package for BLITZ method. The experimental environment is iMac 21.5-inch, macOS Sierra version 10.12.1, Intel Core i5. The implementation and environment will be the same for all experiments unless specified.

2.4.1 Results for Linear Regression

Similar to sequential and dynamic screening algorithms, SAIF can be assembled with different kinds of LASSO solution methods. Shooting algorithm (coordinate minimization) is chosen as the base algorithm in our experiments. Both dynamic screening [38] and SAIF can do feature screening or selection without the help from a heavier parameter solution. We specifically focus on the performance comparison among (1) shooting algorithm without screening (No Scr.), (2) shooting algorithm with dynamic screening [38] (Dyn. Scr), (3) Working set method BLITZ [49] (BLITZ), and (4) shooting algorithm with SAIF screening (SAIF). All of these are safe methods for LASSO problems.

2.4.1.1 Simulation Study

First, we simulate the data sets with $n = 100$ samples and $p = 5,000$ features according to a linear model $\mathbf{y} = X\beta + \epsilon$, where each column of X is a vector with random values uniformly sampled from the interval $[-10, 10]$, and $\epsilon \sim N(0, 1)$. For the linear coefficients β, 20% entries ($0.2p$) are randomly set to the values in $[1, -1]$, and the rest ($0.8p$) to zero. For this data set, we can derive $\lambda_{max} = 2.183 \times 10^4$. The first plot in Figure 2.2 illustrates the running time for different methods in the logarithmic time scale at $\lambda = 20$, 100, and $1,000$. We can see that, SAIF takes much less time than the other methods to reach the optimal solutions with given duality gaps. The results also show that SAIF is more efficient to the feature dimension compared with existing safe methods when the model hyper-parameter is small.

2.4.1.2 Breast Cancer Data

Breast cancer data set consists of gene expression data of 8,141 genes for 78 metastatic and 217 non-metastatic breast cancer patients from the study introduced in [54]. In this set of experiments, the metastatic samples are labeled as 1 and non-metastatic as -1 as the

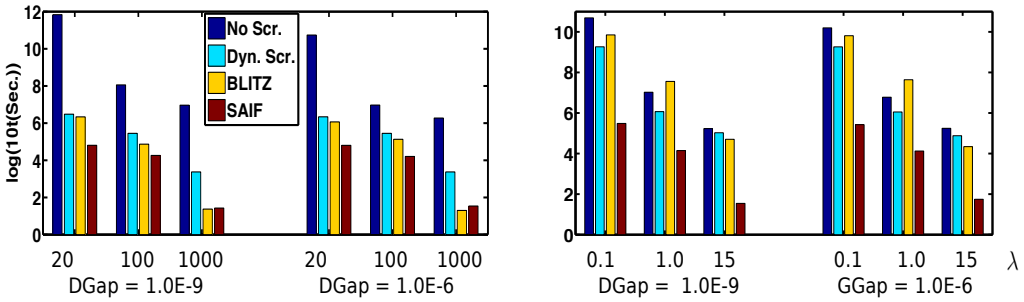


Figure 2.2: Running time comparison on simulation (left) and breast cancer (right).

output of the LASSO linear regression problem. The right plot in Figure 2.2 compares the running time for three different methods at different λ's. Again, SAIF takes the least computation time for different duality gaps.

We further investigate the size of the active set along with the optimization iterations for dynamic screening and SAIF in Figure 2.3-a,c), with $\lambda = 0.1$ and 5. We can see that SAIF starts from a small active feature set and gradually increase its size with time, while dynamic screening starts from the whole feature set and takes longer time to reach the point with screening power. Figure 2.3-c,d) illustrate the change of the dual objective function values $D(\theta_t)$ for SAIF during the optimization procedure. With the active feature set size

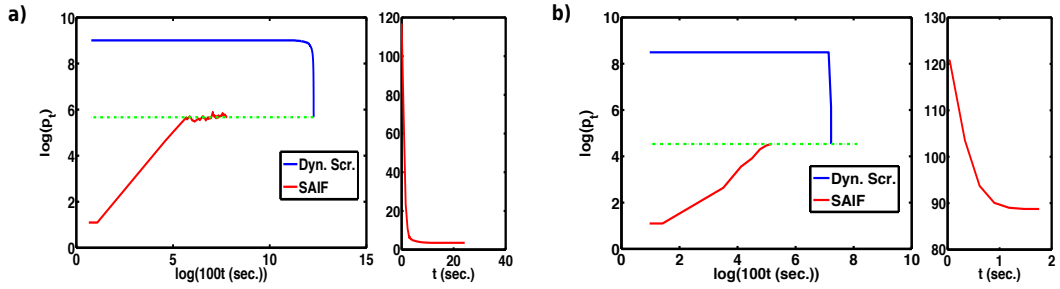


Figure 2.3: a,c) Active feature set size at different time points for breast cancer data with $\lambda = 0.1$ and 5, respectively. Green dotted lines indicate the optimal feature set size. b,d) The corresponding $D(\theta_t)$ value changes with different time points during SAIF optimization at $\lambda = 0.1$ and 5, respectively.

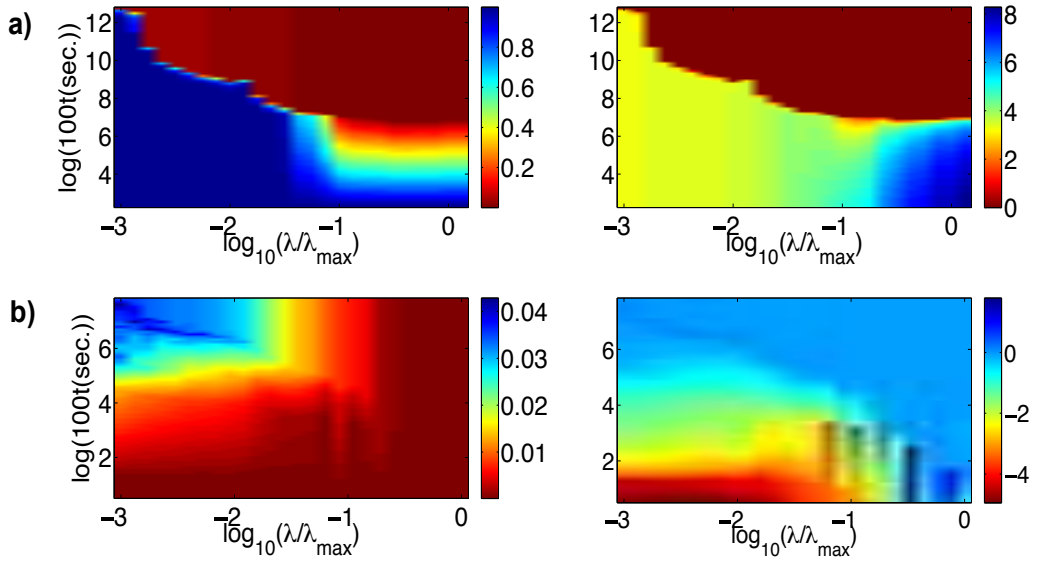


Figure 2.4: $\frac{p_t}{p}$ (left) and $\log(\frac{p_t}{p'})$ (right) as functions of $\log_{10}\frac{\lambda}{\lambda_{max}}$ (x-axis) and $\log(100 \times t(sec.))$ (y-axis) for a) dynamic screening, and b) SAIF on breast cancer data.

increasing, $D(\theta_t)$ decreases and finally converges to a steady value $D(\theta^*)$, indicating the algorithm obtains the optimal solutions to the original LASSO problems.

Let p_t be the feature number at iteration step t for SAIF or dynamic screening. The left column in Figure 2.4 shows the change of $\frac{p_t}{p}$ with respect to the regularization penalty ($\log_{10}\left(\frac{\lambda}{\lambda_{max}}\right)$ on x-axis) and the optimization time ($\log(100 \times t(sec.))$ on y-axis). Similarly, we plot the change of $\log(\frac{p_t}{p'})$, where p' is the corresponding optimal active feature size in the right column of Figure 2.4. From Figure 2.4, it is clear that dynamic screening always takes longer time to reach the optimal active feature set size, especially when λ is small. Before reaching the point with screening power, the active feature set size is almost p. While the active feature set size for SAIF grows gradually from a small set. Due to the small active set size for the starting iterations, SAIF can more efficiently reach the optimal active set size with much shorter running time. All of these results confirms the theoretical complexity analysis for dynamic screening and SAIF. Furthermore, both Figures 2.3

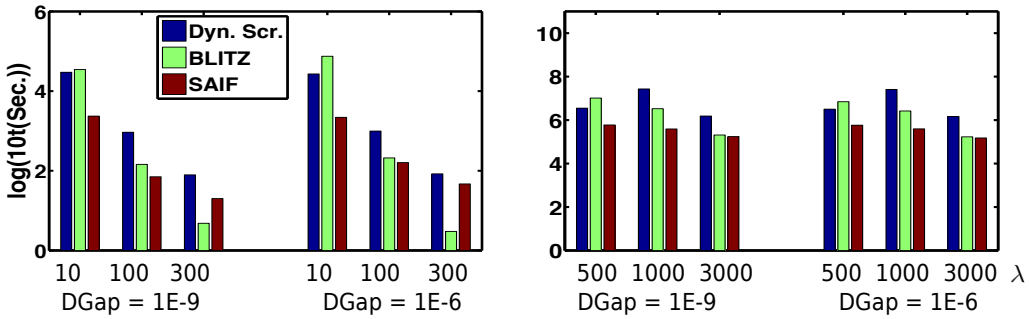


Figure 2.5: Running time comparison on USPS (left) and Gisette (right) data sets.

and 2.4 illustrate that SAIF is more scalable than the existing methods as it always starts from a very small active set and iteratively focuses on a small subset of the features.

2.4.2 Results for Logistic Regression

We evaluate the proposed algorithms for sparse logistic regression with two data sets, Gisette and USPS, from LibSVM [55] Website. The Gisette data set has 5,000 features and 6,000 samples; there are 256 features, 7,291 samples, and 10 labels in the USPS data set, and we categorize the label values large than 4 as positive, and negative otherwise. The λ_{max} is 932,575 and 992, respectively. Figure 2.5 gives the running time at different λ values for dynamic screening, BLITZ, and SAIF. Though due to the implementation issue, BLITZ may achieve comparable performance when the active set is very small, SAIF continuously take less computation at different λ values for both data sets. SAIF can achieve more efficiency for both linear and logistic regression compared with existing safe methods.

2.4.3 Comparison with Sequential Screening and Homotopy Methods

With a sequence of decreasing λ values, SAIF can be further improved with the warm start strategy. Given the simulation and the breast cancer data sets in Section 5.1, a decreasing sequence of λ values are evenly sampled from the logarithmic scale of the range

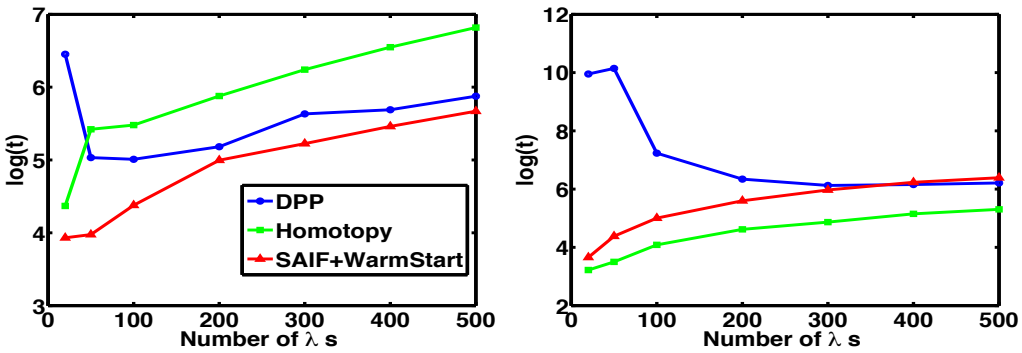


Figure 2.6: Running time for different methods with different number of λ values on simulation (left) and breast cancer (right) data sets.

$[0.001\lambda_{max}, \lambda_{max}]$. The plots in Figure 2.6 present the running time for DPP [11], the homotopy method [48], and SAIF with a different number of λ values on both of data sets. In this set of study, we set the stopping criteria with the duality gap 1.0E-6 for all of the algorithms to achieve fair comparison. The results show that SAIF takes much less time than the DPP method especially when the number of λ is small. With breast cancer data set, the homotopy method can achieve the least computation cost; however, in the result for simulation data, the homotopy method losses its advantages. More critically, the homotopy methods is not safe. Table 2.1 gives the average (Avg.) and standard derivation (Std.) for recall (Rec.) and precision (Prec.) regarding the active features recovered by the homotopy method [48]. According to the recall results, the homotopy method always miss some of the active features at different number of λ values. Furthermore, the homotopy method lead to the inclusion of inactive features into the final solution as evidenced in Table 1 that the precision cannot reach 1 at different numbers of λ values. While our SAIF has the safe guarantee, the recall and precision metrics regarding active features recovered by SAIF are always one. Clearly, the unsafe strategies employed by homotopy methods do not always reduce computation, and the employed inactive features may lead to larger CPU time consumption as shown in the left plot in Figure 2.6.

Table 2.1: Recall and precision for active features recovered by homotopy method at different numbers of λ values.

Num. of λ values	Rec. Avg.	Rec. Std	Prec. Avg.	Prec. Std
20	0.896	0.097	0.972	0.032
50	0.912	0.075	0.982	0.017
100	0.911	0.079	0.979	0.021
200	0.926	0.061	0.974	0.068
300	0.927	0.060	0.969	0.093
400	0.929	0.059	0.971	0.087
500	0.929	0.058	0.976	0.060

2.5 Conclusions

In this chapter, we have developed a novel feature selection method for LASSO–SAIF. From the experimental results, SAIF can achieve improved efficiency compared with existing methods. SAIF has the potential to scale up for data sets with high dimensional features due to its incremental property. Further more, theoretical analysis reveals the safety guarantee and low algorithm complexity of the proposed method. SAIF provides us with a new direction for scaling up sparse learning. Given a data set with extremely high feature dimension, SAIF can be further improved with the multi-level active set and remaining set schema. Furthermore, SAIF can be potentially extended to group LASSO [56] and other sparse models.

3. SAFE FEATURE SCREENING FOR GENERALIZED LASSO

Chapter 2 focuses on scaling up sparse models regularized with L_1 norm with the assumption that model parameters are independent with each other. However, real world data usually contains much complicate structures, and people usually impose these structural knowledge with Fused LASSO, Group LASSO, and Generalized LASSO (GL) into models. A bunch of algorithms and screen methods have been developed for Fused LASSO, Group LASSO. But solving GL problems is challenging, particularly when analyzing many features with a complex interacting structure. Existing methods are mostly devoted to special cases of GL problems with special structures for feature interactions, such as chains or trees. Developing screening rules, particularly, safe screening rules to remove or aggregate features with general interaction structures, calls for a very different screening approach for GL problems. We propose two approaches to tackle this challenge. Firstly, we develop a sequentially screening method for GL. We formulate the GL screening problem as a bound estimation problem in a large linear inequality system when solving them in the dual space. We propose a novel bound propagation algorithm for efficient safe screening for general GL problems, which can be further enhanced by developing novel transformation methods that can effectively decouple interactions among features. Secondly, we show that GL problem with tree structures can be scaled up with SAIF. Experiments on real-world data demonstrate the effectiveness of the proposed screening methods.

3.1 Introduction

Sparse and structured sparse regularization, such as LASSO [16], Fused LASSO [17, 57], and Graph LASSO [34, 58, 59], provide effective tools to incorporate feature sparsity and structure prior knowledge to classification and regression problems when involved features have complex interactions. Such Generalized LASSO (GL) [34, 60, 61, 62] problems

can be summarized by the following optimization formulation:

$$\min_{\beta} \mathbf{f}(X, \mathbf{y}; \beta) + \lambda ||D\beta||_1, \tag{3.1}$$

in which the loss function $\mathbf{f}(\cdot)$ can have different functional forms such as the squared loss function for linear regression, 0/1 loss for logistic regression, hinge loss, and other convex formulations to characterize the prediction performance and guide the learning of functional relationships from observed features X to outcome responses $\mathbf{y}$. The operation matrix D captures structural relationships among features. With different D, we can impose different regularization formulations for learning, such as Fused LASSO, Generalized Fused LASSO (GFL), sparse Generalized Fused LASSO (SGFL), trend filtering, and graph OSCAR (Octagonal Shrinkage and Clustering Algorithm for Regression).

With the data volume and feature dimension growing in an astounding speed, directly solving such sparse and structured sparse problems is challenging. Efficient methods and software packages such as SLEP [63] and MALSAR [64] have been developed to tackle a range of sparse and structured sparse learning problems. The dual path method proposed by [34] can sequentially compute the solutions for all of the valid regularization penalty parameter λ's. This method requires to compute the inversion of the feature matrix, which makes it difficult to scale up to large data sets. The method presented in [57] tries to solve the Sparse Generalized Fused LASSO problem with submodular optimization, but their algorithm cannot be applied to problems with any arbitrary D. Moreover, they did not compare their methods with standard convex optimization solvers such as CVX on large data sets.

As discussed in the introduction chapter, recently, there has been a very exciting discovery that it is possible to screen many features before the use of any optimization solver for learning by developing efficient screening rules [9, 8, 10, 65, 11, 21, 35]. Some of these

derived screening rules are proved to be safe, which means the features that are screened will definitely be inactive or redundant in the actual optimization formulations for the corresponding learning problems [10, 11, 65]. Typical advancements along this direction include the screening methods for LASSO [10, 65, 11], Group LASSO [21], and Fused LASSO [35]. However, none of the existing screening methods can be directly applied to Generalized LASSO (GL) problems because of the complex structure of the operation matrix D in (3.1) when capturing complex interactions among high-dimensional features. Due to the arbitrary and often complex topology than the 1D-chain in Fused LASSO or the tree structure in Tree Group LASSO, it is difficult to transform the GL problems into a form so that we can easily follow LASSO screening strategies as in Group LASSO [11, 36] or 1D-chain Fused LASSO [35] screening approaches. This imposes a significant challenge that calls for a very different screening approach for GL in (3.1) from the existing ones. In the following sections, we first develop a sequential screening method that can apply to more general structure cases. Then we propose an active selection method based on the idea of SAIF for tree Fused LASSO.

3.2 Dual of Generalized LASSO

Assume that we have a data set $X \in R^{n\times p}$ with n data samples and p features; $\mathbf{y}$ is the corresponding outcome or sample label vector, and the entry value of $\mathbf{y}$ can be real, integer, or binary. In this chapter, we focus on the following Generalized LASSO (GL) problem:

$$P: \quad \min_{\beta} \sum_{i}^{n} f(g_{i\bullet}\beta) + \lambda||D\beta||_1. \tag{3.2}$$

Here, $g_{i\bullet}$ is the ith row of a matrix G, which is a general matrix function of the outcome $\mathbf{y}$ and the data sample X; D captures the feature interactive relationships and is typically a sparse matrix; $f(\cdot)$ is a convex loss function, such as the squared or logistic loss function.

We also assume $\mathbf{f}(\beta) = \sum_{i=1}^{n} f(g_{i\bullet}\beta)$. For example, the GL regression problem can be written as the following optimization problem:

$$\min_{\beta} \frac{1}{2}||\mathbf{y} - X\beta||_2^2 + \lambda||D\beta||_1, \tag{3.3}$$

in which G is simply the design matrix X. As examples, for LASSO, D is an identity matrix; and for 1D-chain Fused LASSO [17, 35], D can be written as follows:

$$D = \begin{bmatrix} 1 & -1 & 0 & ... & 0 & 0 \\ 0 & 1 & -1 & ... & 0 & 0 \\ & & & ... & & \\ 0 & 0 & 0 & ... & 1 & -1 \end{bmatrix}. \tag{3.4}$$

To facilitate the derivation of the screening rules for any GL problem P, we study its dual problem. Let f^* be the conjugate function of f. We can derive the dual problem of (3.63) based on the following theorem.

Theorem 1 *A dual form of* (3.63) *is given by*

$$D : \min_{\theta \in \Omega_\lambda} \mathbf{f}^*(\theta) = \sum_{i=1}^{n} f^*(\theta_i),\ \Omega_\lambda = \{\theta : G^T\theta = \lambda D^T u, ||u||_\infty \leq 1\}. \tag{3.5}$$

The primal and dual relationship is $f'(g_{i\bullet}^T\beta) = \theta_i$ with $f'(z)$ denoting the derivative of $f(z)$ with respect to z.

Proof:

We here provide the derivation of the dual problem for Generalized LASSO (GL). For the original problem,

$$\min_{\beta} \sum_{i=1}^{n} f(g_{i\bullet}^T\beta) + \lambda||D\beta||_1. \tag{3.6}$$

Table 3.1: Dual forms of different loss functions in (3.63)

	$f(z)$	$g_{i\bullet}$	$f^*(\theta_i)$
Linear Regression	$\frac{1}{2}\|\|y_i - z\|\|_2^2$	$x_{i\bullet}$	$\frac{1}{2}\|\|y_i + \theta_i\|\|^2 - \frac{1}{2}y_i^2$
Logistic Regression	$\log(1+\exp(z)) - zy_i$	$x_{i\bullet}$	$(y_i+\theta_i)\log(y_i+\theta_i)$ $+(1-y_i-\theta_i)\log(1-y_i-\theta_i)$
Multinomial Regression	$\log(\sum_k \exp(z_k)) - \sum_k z_k y_{i(k)}$	$x_{i\bullet}$	$\sum_k(\theta_{i(k)}+y_{i(k)})\log(\theta_{i(k)}+y_{i(k)})$

With θ as the corresponding Lagrangian multiplier, we can write the Lagrangian as follows

$$L(\beta, z, \lambda; \theta) = \sum_{i=1}^{n} f(z_i) + \lambda ||D\beta||_1 + \theta^T (G\beta - z). \tag{3.7}$$

Let

$$f_\beta = \lambda ||D\beta||_1 + \theta^T G\beta, \;\; f_{z_i} = f(z_i) - \theta_i z_i. \tag{3.8}$$

To get the dual form, we need to minimize f_β and f_z. Since

$$\partial_\beta f_\beta = G^T\theta + \lambda D^T u, \tag{3.9}$$

where $u \in sign(D\beta)$, and $||u||_\infty \leq 1, u^T D\beta = ||D\beta||_1$. To minimize f_β, we have

$$0 \in \partial_\beta f_\beta \Rightarrow \exists u, -G^T\theta = \lambda D^T u, \;\; \Rightarrow \;\; \min_\beta f_\beta = (\lambda u^T D + \theta^T G)\beta = 0. \tag{3.10}$$

We also have

$$0 = \partial_{z_i} f_{z_i} \Rightarrow \theta_i = f'(z_i) \tag{3.11}$$

$$\min_{z_i} f_{z_i} = \min_{z_i} f(z_i) - \theta_i z_i = \min_{z_i} -(\theta_i z_i - f(z_i)) \tag{3.12}$$

$$= -\max_{z_i}(\theta_i z_i - f(z_i)) \stackrel{\Delta}{=} -f^*(\theta_i) \tag{3.13}$$

With (3.10) and (3.13), we can have the dual objective function as follows:

$$\max_\theta L(\theta) = \max_\theta -\sum_{i=1}^{n} f^*(\theta_i) \tag{3.14}$$

With the constraints on θ, the dual problem is

$$\max_{\theta} L(\theta) = \max_{\theta} -\sum_{i=1}^{n} f^*(\theta_i) \tag{3.15}$$

$$s.t. -G^T\theta = \lambda D^T u \qquad ||u||_\infty \leq 1. \tag{3.16}$$

From (3.11), the primal and dual variables satisfy the following equation

$$f'(g_{i\bullet}\beta) = \theta_i. \tag{3.17}$$

As the feasible region for u is symmetric, we can move the negative sign in (3.16) to right side and the above dual form can be rewritten as

$$\min_{\theta} \sum_{i=1}^{n} f^*(\theta_i) \tag{3.18}$$

$$s.t.\ \ G^T\theta = \lambda D^T u, \quad ||u||_\infty \leq 1. \tag{3.19}$$

And the primal and dual relationship is

$$f'(g_{i\bullet}\beta) = \theta_i. \tag{3.20}$$

In the above theorem, θ denotes the dual variables; u denotes the sub-gradient vector of $||D\beta||_1$ with respect to $D\beta$, and u can be considered as an auxiliary vector in the dual form. With Theorem 1, we can derive the dual forms of many GL problems with different convex loss functions. For example, the dual problem of GL regression (3.3) can be written as:

$$\min_{\theta} \left\{ \frac{1}{2}||\theta + \mathbf{y}||_2^2 - \frac{1}{2}||\mathbf{y}||_2^2,\ s.t.\, X^T\theta = \lambda D^T u, ||u||_\infty \leq 1 \right\} \tag{3.21}$$

with $-\mathbf{y}+X\beta=\theta$ as the primal-dual relationship. Table 3.1 gives the dual forms of some standard loss functions [38] in GL learning.

The dual variables have the following properties:

Theorem 2 *For Generalized LASSO problems* (3.63)*:*

a) If θ_0^* and θ^* are the optimal solutions to the dual problem (3.5) *at λ_0 and λ, then we have*

$$\langle \mathbf{f}'^*(\theta_0^*) - \frac{\lambda}{\lambda_0}\mathbf{f}'^*(\theta^*), \theta^* - \frac{\lambda}{\lambda_0}\theta_0^* \rangle \geq 0,$$

and

$$\langle \mathbf{f}'^*(\theta_0^*) - \mathbf{f}'^*(\theta^*), \frac{\theta^*}{\lambda} - \frac{\theta_0^*}{\lambda_0} \rangle \geq 0.$$

b) If $\mathbf{f}^*$ is α-strongly convex, and θ_0^* and θ^* are the optimal solutions to the dual problems at λ_0 and λ with $\lambda < \lambda_0$, then

$$||\theta^* - \theta_0^*||_2^2 \leq \frac{2}{\alpha}\left[\mathbf{f}^*(\frac{\lambda}{\lambda_0}\theta_0^*) - \mathbf{f}^*(\theta_0^*) + (1-\frac{\lambda}{\lambda_0})\langle \mathbf{f}'^*(\theta_0^*), \theta_0^* \rangle\right].$$

c) For GL linear regression problems with $\lambda < \lambda_0 < \lambda_{max}$, and θ^* and θ_0^* are the optimal dual solutions at λ and λ_0, respectively, we have

$$||\theta^* - \frac{\lambda}{\lambda_0}\theta_0^* + \frac{1}{2}\mathbf{v}||_2 \leq \frac{1}{2}||\mathbf{v}||_2, \tag{3.22}$$

where

$$\mathbf{v} = \lambda(\mathbf{v}_2 - \frac{\langle \mathbf{v}_1, \mathbf{v}_2 \rangle}{||\mathbf{v}_1||_2^2}\mathbf{v}_1), \mathbf{v}_1 = \frac{\mathbf{y}}{\lambda_0} + \frac{\theta_0^*}{\lambda_0}, \mathbf{v}_2 = \frac{\mathbf{y}}{\lambda} + \frac{\theta_0^*}{\lambda_0}.$$

Proof:

a) We transform the dual form into the following form

$$D_2 : \min_{\hat{\theta} \in \Omega} \mathbf{f}^*(-\lambda\theta') = \sum_{i=1}^{n} f^*(-\lambda\theta'_i), \tag{3.23}$$

$$\Omega = \{\theta' : G^T\theta' = D^T u, ||u||_\infty \leq 1\}. \tag{3.24}$$

With a given λ, the solution relationship between D_2 and the dual form is $\theta'^* = -\frac{\theta^*}{\lambda}$. We can see that with different λ's, the corresponding optimization problems still have the same feasible region. According to [66], for a constrained optimization problem, $\min_{\mathbf{x}\in\Phi} \mathbf{h}(\mathbf{x})$, with Φ being convex and closed and $\mathbf{h}(\cdot)$ being convex and differentiable, we have the following relationship for an optimal solution $\mathbf{x}^*$:

$$\langle \mathbf{h}'(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq 0, \quad \forall \mathbf{x} \in \Phi. \tag{3.25}$$

Let θ'^* and θ'^*_0 be the optimal solution of D_2 at λ and λ_0, we have

$$\langle -\lambda \mathbf{f}'^*(-\lambda\theta'^*), \theta'^*_0 - \theta'^* \rangle \geq 0, \quad \langle -\lambda_0 \mathbf{f}'^*(-\lambda_0\theta'^*_0), \theta'^* - \theta'^*_0 \rangle \geq 0. \tag{3.26}$$

Thus we have

$$\langle \lambda_0 \mathbf{f}'^*(-\lambda_0\theta'^*_0) - \lambda \mathbf{f}'^*(-\lambda\theta'^*), \theta'^*_0 - \theta'^* \rangle \geq 0, \text{ and } \langle \mathbf{f}'^*(-\lambda_0\theta'^*_0) - \mathbf{f}'^*(-\lambda\theta'^*), \theta'^*_0 - \theta'^* \rangle \geq 0,$$

which lead to

$$\langle \mathbf{f}'^*(\theta^*_0) - \frac{\lambda}{\lambda_0}\mathbf{f}'^*(\theta^*), \theta^* - \frac{\lambda}{\lambda_0}\theta^*_0 \rangle \geq 0, \text{ and } \langle \mathbf{f}'^*(\theta^*_0) - \mathbf{f}'^*(\theta^*), \frac{\theta^*}{\lambda} - \frac{\theta^*_0}{\lambda_0} \rangle \geq 0,$$

respectively.

b) We use Ω_λ to represent the feasible region of the dual problem at λ. As $\lambda < \lambda_0$, we

can easily get $\Omega_\lambda \subset \Omega_{\lambda_0}$.

$\mathbf{f}^*(\cdot)$ is a α-strongly convex function, we have

$$||\theta^* - \theta_0^*||_2^2 \leq \frac{2}{\alpha}\left[\mathbf{f}^*(\theta^*) - \mathbf{f}^*(\theta_0^*) - \langle \mathbf{f}'^*(\theta_0^*), \theta^* - \theta_0^* \rangle\right]. \tag{3.27}$$

Let $\tilde{\theta}^* = \frac{\lambda}{\lambda_0}\theta_0^*$. We have $G^T\tilde{\theta}^* = \frac{\lambda}{\lambda_0}G^T\theta_0^* = \lambda D^T u_0^*$. As $||u_0^*||_\infty \leq 1$, we have $\tilde{\theta}^* \in \Omega_\lambda = \{\theta \mid G^T\theta = \lambda D^T u, ||u||_\infty \leq 1\}$, and

$$\mathbf{f}^*(\theta^*) = \min_{\theta \in \Omega_\lambda} \mathbf{f}^*(\theta) \leq \mathbf{f}^*(\tilde{\theta^*}) = \mathbf{f}^*(\frac{\lambda}{\lambda_0}\theta_0^*). \tag{3.28}$$

As in the proof of a), we convert the dual problem into the similar form D_2. With (3.25), we have

$$\langle -\lambda_0 \mathbf{f}'(\theta_0^*), \theta - (-\frac{\theta_0^*}{\lambda_0}) \rangle \geq 0, \ \forall \theta \in \Phi. \tag{3.29}$$

Here $\Omega = \left\{\theta \mid G^T\theta = D^T u, ||u||_\infty \leq 1\right\}$. As θ^* is the optimal dual solution at λ, thus $-\frac{\theta^*}{\lambda} \in \Omega$. Then we have

$$\langle -\lambda_0 \mathbf{f}'(\theta_0^*), -\frac{\theta^*}{\lambda} - (-\frac{\theta_0^*}{\lambda_0}) \rangle \geq 0 \implies \langle -\mathbf{f}'(\theta_0^*), -\frac{\theta^*}{\lambda} - (-\frac{\theta_0^*}{\lambda_0}) \rangle \geq 0 \tag{3.30}$$

$$\implies -\langle \mathbf{f}'(\theta_0^*), \theta^* \rangle \leq -\langle \mathbf{f}'(\theta_0^*), \frac{\lambda}{\lambda_0}\theta_0^* \rangle. \tag{3.31}$$

Plugging (3.28) and (3.31) into (3.27), we get

$$||\theta^* - \theta_0^*||_2^2 \leq \frac{2}{\alpha}\left[\mathbf{f}^*(\frac{\lambda}{\lambda_0}\theta_0^*) - \mathbf{f}^*(\theta_0^*) + (1 - \frac{\lambda}{\lambda_0})\langle \mathbf{f}'^*(\theta_0^*), \theta_0^* \rangle\right].$$

c) According to Theorem 1, the dual form for linear regression is

$$\min_\theta \left\{\frac{1}{2}||\theta + \mathbf{y}||_2^2 - \frac{1}{2}||\mathbf{y}||_2^2 : X^T\theta = \lambda D^T u, ||u||_\infty \leq 1\right\}, \tag{3.32}$$

which can be reformulated as:

$$\min_{\theta'} \left\{ \frac{\lambda^2}{2}||\theta' - \frac{\mathbf{y}}{\lambda}||_2^2 - \frac{1}{2}||\mathbf{y}||_2^2 : X^T\theta' = D^T u, ||u||_\infty \leq 1 \right\}. \tag{3.33}$$

We can see that at the same λ, the optimal solution to (3.32) and (3.33) have the following relationship:

$$\theta'^* = -\frac{\theta^*}{\lambda}. \tag{3.34}$$

According to [11], when $\lambda < \lambda_0 < \lambda_{max}$, all of the projection properties used in Dual Polytope Projection (DPP) and enhanced DPP still hold regarding to the objective of (3.33). Let $\mathbf{v}_1 = \frac{\mathbf{y}}{\lambda_0} - \theta_0'^*, \mathbf{v}_2 = \frac{\mathbf{y}}{\lambda} - \theta_0'^*$. With Theorem 15 in [11], we have

$$\left|\left|\theta_\lambda'^* - \left(\theta_{\lambda_0}'^* + \frac{1}{2}\mathbf{v}_2^\perp\right)\right|\right|_2 \leq \frac{1}{2}||\mathbf{v}_2^\perp||_2,$$

where $\mathbf{v}_2^\perp = \mathbf{v}_2 - \frac{\langle \mathbf{v}_1, \mathbf{v}_2 \rangle}{||\mathbf{v}_1||_2^2}$. With the optimal solution relationship (3.34), we have

$$\left|\left|\theta^* - \frac{\lambda}{\lambda_0}\theta_0^* + \frac{1}{2}\mathbf{v}\right|\right|_2 \leq \frac{1}{2}||\mathbf{v}||_2, \tag{3.35}$$

where

$$\mathbf{v} = \lambda\left(\mathbf{v}_2 - \frac{\langle \mathbf{v}_1, \mathbf{v}_2 \rangle}{||\mathbf{v}_1||_2^2}\mathbf{v}_1\right), \mathbf{v}_1 = \frac{\mathbf{y}}{\lambda_0} + \frac{\theta_0^*}{\lambda_0}, \mathbf{v}_2 = \frac{\mathbf{y}}{\lambda} + \frac{\theta_0^*}{\lambda_0}.$$

Note that Theorem 2 is generic for a wide range of loss functions. For instance, for logistic regression, $\mathbf{f}^*(\theta_i) = (y_i+\theta_i)\log(y_i+\theta_i)+(1-y_i-\theta_i)\log(1-y_i-\theta_i)$, and $\mathbf{f}''^*(\theta_i) = \frac{1}{(y_i+\theta_i)(1-y_i-\theta_i)} \geq 4$, so $\mathbf{f}(\cdot)$ is 4-strongly convex, and we can directly use Theorem 2-b) to estimate the region for θ^* at λ. For linear regression, Theorem 2-c) usually gives a tighter bound. There are many existing methods for different loss functions with the classic LASSO penalty [10, 11, 67, 38]. LASSO screening [67] derives their screening rules in a similar way as in Theorem 2-a). It is possible to derive tighter bounds for dual variables

given some properties of the loss functions.

The intersection of the constraint regions from these inequalities in Theorem 2 can give us tighter bound estimates for the optimal dual solution at λ. In the next section, we show how to derive SAIF feature selection rule based on the dual form. In the third section, we show how to derive the screening rules for GL problems based on the corresponding constraint regions.

3.3 Sequential Screening Rules for Generalized LASSO (GL)

In this section, we follow the sequential screening approach survey in Chapter 1, and propose a screening method for GL. The main contributions of this method include: a) We show that the safe GL screening problem can be formulated as a bound estimation problem constrained by a linear inequality system derived based on the equivalent dual problem. We also provide effective dual variable range estimation approaches that give the initial upper and lower bounds of the linear system for a broad range of loss functions; b) A novel bound propagation algorithm is developed to efficiently approximate the feasible solution space for the linear inequality system to derive tight bound estimates; c) We show that the efficiency of our bound propagation algorithm can be further improved by our graph transformation methods; d) The proposed propagation and transformation methods can also be applicable with dynamic screening [38, 39, 40], which further provides an efficient way to start the screening process when the desirable regularization parameter λ is difficult to estimate. The experimental results on synthetic and real-world data sets have shown clear advantages of incorporating our safe screening method in GL learning.

3.3.1 Derivation of Safe Screening Rules

A safe screening rule is to identify the items that take zero values within the L_1 regularization term in the primal problem (3.63) solution at any given λ. The kth L_1 regularization item is **trivial** if it is zero in the solution of the given λ, which corresponds to $D_{k\bullet}\beta^* = 0$

in the matrix form. In what follows, we show that the underlying computational task of deriving safe screening rules involves the estimation of the ranges of the sub-gradient vector for $||D\beta^*||_1$ with respect to each entry of $D\beta^*$ at λ, denoted as $u^*(\lambda)$. u_k^* is the kth entry of $u^*(\lambda)$. To see that, note that $D_{k\bullet}\beta^* = 0$, if and only if $|u_k^*| < 1$ by the definition of the sub-gradient. Thus, once we have a range set U for the vector $u^*(\lambda)$, we can derive the following screening rule for each entry u_k in u, where u_k corresponds to the kth L_1 item:

$$\sup_{u \in U} |u_k| < 1 \Rightarrow D_{k\bullet}\beta^* = 0 \ \ (\textbf{R1}). \tag{3.36}$$

Therefore, to decide whether the kth item is trivial or not at the given λ, the task is to estimate the range of u_k with U. From the dual form in (3.5), the range of the sub-gradient vector $u^*(\lambda)$ is constrained by $G\theta^*$. Thus if we can estimate the range of θ^*, we can determine the range of $G\theta^*$, then determine the range of $D^T u^*(\lambda)$, which will ultimately lead to the estimation of the range of $u^*(\lambda)$. Note that, in existing screening methods such as those for LASSO, without a complex structure for D, it is straightforward to translate the estimation of the range of θ^* to $u^*(\lambda)$. Thus, LASSO is a special case of our problem, for which $\theta^*(\lambda)$ can be bounded by a ball: $B(\tau, r) : ||\theta^*(\lambda) - \tau||_2 \leq r$ as in the ball test for LASSO screening [10, 11, 38, 39], with τ being the center and r the radius.

Let $\theta^*(\lambda)$ denote the optimal dual variable at a given penalty parameter λ. We follow the results in Theorem 2 to derive the ball region for GL problems. Let $\theta^*(\lambda) = \tau + \rho$. We have $g_{\bullet i}^T\theta^*(\lambda) = g_{\bullet i}^T\tau + g_{\bullet i}^T\rho$. As $||\rho||_2 \leq r$, $g_{\bullet i}^T\theta^*(\lambda)$ can be bounded as follows:

$$g_{\bullet i}^T\tau - r||g_{\bullet i}||_2 \leq g_{\bullet i}^T\theta^*(\lambda) \leq g_{\bullet i}^T\tau + r||g_{\bullet i}||_2, \tag{3.37}$$

which gives the upper and lower bounds for $G^T\theta^*(\lambda)$: $L \leq \frac{1}{\lambda}G^T\theta^*(\lambda) \leq H$. The ith

entries in L and H are given by $L_i = \frac{1}{\lambda}(g_{\bullet i}^T\tau - r||g_{\bullet i}||_2)$, and $H_i = \frac{1}{\lambda}(g_{\bullet i}^T\tau + r||g_{\bullet i}||_2)$. From (3.5), we can see

$$L \leq D^T u^*(\lambda) \leq H. \tag{3.38}$$

Since $u^*(\lambda)$ is a sub-gradient vector, we define the inequality set for $u^*(\lambda)$ as $U = \{u : L \leq D^T u \leq H, -\mathbf{1} \leq u \leq \mathbf{1}\}$, in which L and H are screening bounds. Estimating the bounds for each u_k^* subject to the constraint, $u^*(\lambda) \in U$, is a challenging computational problem.

Before we tackle the problem by introducing a novel bound propagation algorithm in Section 3, we first establish that the derived screening rule **R1** is safe for aggregating variables to reduce the problem size. Note that to apply this screening rule **R1**, we need to start with a given λ_0, which can be any non-negative value. Given a sequence of descending λ's, we can sequentially screen and aggregate features so that the computational cost is reduced for all λ's.

3.3.2 Safe Feature Elimination and Aggregation

If $D_{k\bullet}$ has only one non-zero entry, e.g., d_{ki}, $|u_k^*| < 1$ corresponds $\beta_i^* = 0$. For this case, we define an **elimination** operator, which removes the column $g_{\bullet i}$ from G and remove the ith column and kth row of D as well.

If $D_{k\bullet}$ has more than one non-zero entries, e.g. d_{ki},d_{kj},...,d_{km}, applying the screening rule **R1** leads to $d_{ki}\beta_i^* + d_{kj}\beta_j^* + ... + d_{km}\beta_m^* = 0$. Thus, we have

$$\beta_i^* = -\frac{d_{kj}}{d_{ki}}\beta_j^* - ... - \frac{d_{km}}{d_{ki}}\beta_m^* = \mathbf{t}_i\beta'^*, \tag{3.39}$$

$$\beta_i^* g_{i\bullet} = \mathbf{t}_i\beta'^* g_{i\bullet}. \tag{3.40}$$

Let $\mathbf{t}_i' = [..., 0, ..., -\frac{d_{kj}}{d_{ki}}, ..., -\frac{d_{km}}{d_{ki}}, ..., 0, ...]_{1\times p}$. Note that $\mathbf{t}_i$ and β'^* are the corresponding

sub-vectors of $\mathbf{t}_i'$ and β^* by removing their corresponding ith entries. For this case, we define an **aggregation** operator:

1. Add the vector $d_{ri}\mathbf{t}_i'$ to the rows $D_{r\bullet}$ with $d_{ri} \neq 0$,
2. Remove the kth row and ith column of D,
3. Update the feature set with $G' = GT_i$, where T_i is a $p \times (p-1)$ matrix with $\mathbf{t}_i$ being the ith row and all of the remaining rows forming a $(p-1)\times(p-1)$ diagonal matrix.

Once we know the range of the sub-gradient vector $u^*(\lambda)$, we can sequentially and safely aggregate the features to reduce the problem size. P' is the reduced-size problem:

$$P': \quad \min_{\beta'} \sum_{i}^{n} f(g_{i\bullet}'\beta') + \lambda||D'\beta'||_1. \tag{3.41}$$

One can reconstruct the original solution for each aggregation operation by the transformation $\beta^* = T_i\beta'^*$. Similarly, for each elimination operation, reconstruction can be done with T_i by inserting one all-zero row to a diagonal matrix. Thus the original solution can be recovered by $\beta^* = T_{i1} \times T_{i2} \times ... \times T_{it}\beta'^* = T\beta'^*$. We can derive T for any reduced problem.

For a fixed λ, let S be the solution set for the original problem P, S' be the optimal solution set for the reduced problem P', and $\tilde{S}'$ be the reconstructed solution set. We have $\beta^* \in S$, $\beta'^* \in S'$, and $\tilde{\beta}'^* \in \tilde{S}'$. Let $P(\tilde{\beta}'^*)$ represent the value of the objective function of P at $\tilde{\beta}'^*$. Similarly, $P'(\beta'^*)$ for P' at β'^*. Assume $\tilde{\beta}'^* = T\beta'^*$, and $\bar{\beta}'^*$ is the reduced solution of β^*. We have the following Theorem to guarantee the equivalence of the problems P and P'.

Theorem 3 *For any Generalized LASSO problem with the penalty parameter λ,*

a) $P'(\bar{\beta}'^*) = P(\beta^*)$; $P'(\beta'^*) = P(\tilde{\beta}'^*)$.

b) The extended optimal solution set of the reduced-size problem P' (3.41) equals to the solution set of the original problem P (3.63).

Proof: a) We first prove $P'(\bar{\beta}'^*) = P(\beta^*)$. For sequential operations, if we can prove at each step the equation holds, then the equation is correct for all operations.

For elimination, we can see that $\mathbf{f}(G'\bar{\beta}'^*) = \mathbf{f}(G\beta^*)$, and $\lambda||D'\bar{\beta}'^*||_1 = \lambda||D\beta^*||_1$. For aggregation, we first prove $\mathbf{f}(G'\bar{\beta}'^*) = \mathbf{f}(G\beta^*)$. As $G'\bar{\beta}'^* = GT\bar{\beta}'^*$, we just need to prove $T\bar{\beta}'^* = \beta^*$. From (3.39) and (3.40), we have $\beta_i = \mathbf{t}_i\beta'$. Therefore, $T\bar{\beta}'^* = \beta^*$. Next we prove $\lambda||D'\bar{\beta}'^*||_1 = \lambda||D\beta^*||_1$. After we expand both sides of the equation, as the aggregation operator replaces β_i^* with $-\frac{d_{kj}}{d_{ki}}\beta_j^* - ... - \frac{d_{km}}{d_{ki}}\beta_m^*$ by (3.39), we can derive $\lambda||D'\bar{\beta}'^*||_1 = \lambda||D\beta^*||_1$.

Now we prove $P'(\beta'^*) = P(\tilde{\beta}'^*)$. As $\tilde{\beta}'^* = T\beta'^*$ and $G' = GT$, we get $\mathbf{f}(G'\beta'^*) = \mathbf{f}(G\tilde{\beta}'^*)$. To prove $\lambda||D'\beta'^*||_1 = \lambda||D\tilde{\beta}'^*||_1$, we need to prove $||D'\beta'^*||_1 = ||DT\beta'^*||_1$. After we insert T into $||D'\beta'^*||_1 = ||DT\beta'^*||_1$, we can see that the right-hand side has one more L_1 term, which is zero, and the remaining terms are exactly the same, which proves Theorem 3-a).

b) $\forall\tilde{\beta}'^* \in \tilde{S}'$, we prove that $\tilde{\beta}'^* \in S$ by contradiction. For $\tilde{\beta}'^*$, we use β'^* to represent the corresponding optimal solution to P'. Let's assume $\tilde{\beta}'^* \notin S$. According to the convexness of the problem, $\exists\bar{\beta}^* \in S$, and $P(\bar{\beta}^*) < P(\tilde{\beta}'^*)$. Let's construct a solution in S' with $\bar{\beta}^*$, i.e., $\bar{\beta}'$, so that $P'(\bar{\beta}'^*) = P(\bar{\beta}) < P(\tilde{\beta}'^*) = P'(\beta'^*)$. This contradicts with the fact that β'^* is in the optimal solution set of P'.

$\forall\bar{\beta}^* \in S$, we prove that $\bar{\beta}^* \in \tilde{S}'$. Similarly, assume $\bar{\beta}^* \notin \tilde{S}'$, then $\exists\tilde{\beta}' \in \tilde{S}'$, and $P(\tilde{\beta}') < P(\bar{\beta}^*)$. This contradicts with the fact that $\bar{\beta}^*$ is in the optimal solution set of P. Hence, we prove that $\forall\bar{\beta}^* \in S$, $\bar{\beta}^* \in \tilde{S}'$.

3.3.3 Bound Propagation for Screening

To apply the screening rule **R1**, we need to estimate the bounds for the entries in $u^*(\lambda)$. Particularly, our objective is to identify as many trivial regularization items as possible. Since an L_1 item is trivial as long as $|u_k^*| < 1$, we need to estimate the upper and lower bounds for u_k as tight as possible, which will increase the chance of finding the items that indeed lead to $|u_k^*| < 1$.

Since $u^*(\lambda) \in U$, the bound estimation for evaluating the screening rule **R1** can be obtained by solving two linear programming (LP) problems:

$$\begin{array}{ll} \min\limits_{u} \ u_i & \max\limits_{u} \ u_i \\ s.t. \ L \leq D^T u \leq H & s.t. \ L \leq D^T u \leq H \\ -\mathbf{1} \leq u \leq \mathbf{1}; & -\mathbf{1} \leq u \leq \mathbf{1}. \end{array} \tag{3.42}$$

Standard simplex and interior point methods can be used to solve these problems. However, it may be very computationally costly as we need to run the LP solver for every u_i twice for the lower and upper bounds. A recent speedup has been proposed to solve similar linear inequality systems [68]. But the algorithm can only identify one of the feasible solutions to an inequality system, which doe not identify bounds tight enough for the efficacy of the proposed screening method to remove as many trivial L_1 items as possible. Inspired by [68], we propose a new bound propagation algorithm to provide an efficient approach for the safe Generalized LASSO screening.

We show the basic idea of the proposed bound propagation algorithm below. Let h_{u_i} and l_{u_i} be the upper and lower bounds of u_i, and h and l are the upper and lower bounds

of u . First, we convert the inequality constraints of u_i in (3.42) in the following form:

$$I_h : -d_{j_1 i} u_{j_1} - d_{j_2 i} u_{j_2} - ... - d_{j_t i} u_{j_t} + H_i \geq 0; \tag{3.43}$$

$$I_l : \ d_{j_1 i} u_{j_1} + d_{j_2 i} u_{j_2} + ... + d_{j_t i} u_{j_t} - L_i \geq 0, \tag{3.44}$$

which will give the bounds of u_i as: $u_i - l_{u_i} \geq 0; \quad -u_i + h_{u_i} \geq 0$. Obviously, the bound estimates of u_i depend on the estimated bounds of other interacting features. We note that the special diagonal structure of D for LASSO screening leads to efficient screening due to its non-interacting features in the regularization term. Since the estimation of the bounds for the variables are interdependent, we design the bound propagation that iteratively updates the bounds of each variable sequentially. We can set the initial values for l_{u_i} and h_{u_i} as 1, which are named as the initial *context* for u_i. Then, in each bound propagation step, we update the bounds for each variable in the above inequalities: l_{u_i} and h_{u_i}, and derive the new bounds of each u_i by variable elimination.

Our procedure can be illustrated using a simple example, when the inequality constraint for the sub-gradient vector u is

$$-2u_1 + 3u_2 - u_3 + 0.4 \geq 0, \tag{3.45}$$

and the current contexts for the bound estimates of u_1, u_2 and u_3 are:

$$(a) \ -u_1 + 1 \geq 0; \qquad (b) \ u_1 + 1 \ \geq 0; \tag{3.46}$$

$$(a) \ -u_2 + 0.1 \geq 0; \qquad (b) \ u_2 + 0.7 \geq 0; \tag{3.47}$$

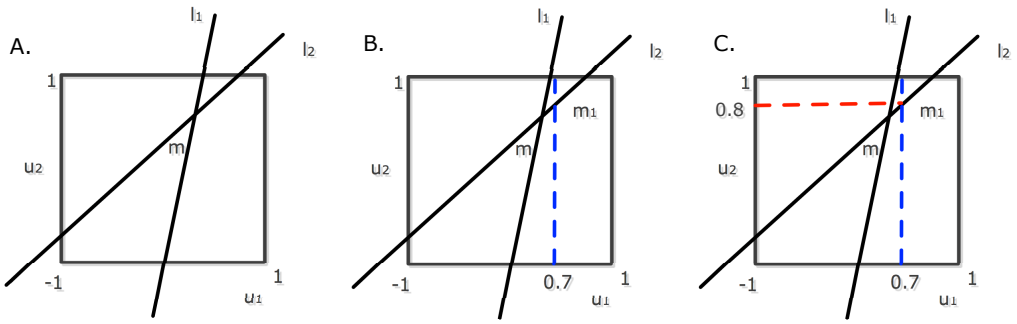
$$(a) \ -u_3 + 0.6 \geq 0; \qquad (b) \ u_3 + 0.8 \geq 0. \tag{3.48}$$

We can lift these inequalities by adding $3\times$ (3.47)(a) and (3.48)(b) to (3.45) to get one

propagated bound for u_1: $-u_1 + 0.75 \geq 0$. As 0.75 is smaller than 1, we take it as the new bound for u_1. Otherwise, we keep the bound unchanged. We can apply the bound propagation in a breadth-first manner to iteratively tighten the estimated bounds. Algorithm 3 provides the details about of the Bound Propagation (BP) procedure, where $\odot$ denotes the element-wise multiplication. Fig. 3.1 provides schematic illustration of BP procedure with inequalities contain two variables.

Data: T^p and T^n, are array lists for positive and negative entries in any column of D; A is the array lists for the indices of non-zero entries in any column of D; Ineuqlity bounds L, H; Initial context bounds for u, l, h
Result: Estimated bounds l and h for all u_i
while l *or* h *is updated* **do**
  for $i = 1$ **to** p **do**
    Let $B_{\max} = T^p[i] \odot h[A[i]] + T^n \odot l[A[i]]$;
     $B_{\min} = T^p[i] \odot l[A[i]] + T^n \odot h[A[i]]$;
    Let $S_{\max} = \sum_j B_{\max}[j]$; $S_{\min} = \sum_j B_{\min}[j]$;
    for $j \in A[i]$ **do**
      if $D[j][i] > 0$ **then**
        $\hat{h}_{u_j} = (-S_{\min} + B_{\min}[j] + H[i])/D[j][i]$;
        $\hat{l}_{u_j} = (-S_{\max} + B_{\max}[j] + L[i])/D[j][i]$;
      else
        $\hat{l}_{u_j} = (-S_{\min} + B_{\min}[j] + H[i])/D[j][i]$;
        $\hat{h}_{u_j} = (-S_{\max} + B_{\max}[j] + L[i])/D[j][i]$;
      end
      $h_{u_j} = \min\{h_{u_j}, \hat{h}_{u_j}\}$;
      $l_{u_j} = \max\{l_{u_j}, \hat{l}_{u_j}\}$
    end
  end
end

Algorithm 3: Bound Propagation (BP).



$u_1 \in [-1, 1]$
$u_2 \in [-1, 1]$

$u_1 \in [-1, 0.7]$
$u_2 \in [-1, 1]$

$u_1 \in [-1, 0.7]$
$u_2 \in [-1, 0.8]$

Figure 3.1: Schematic illustration of bound propagation algorithm. In the figures l_1 and l_2 are two lines corresponding two inequalities of u_1 and u_2. (A) Initial context for u_1 and u_2 as the illustrated box. (B) Upper bound for u_1 is updated to 0.7 based on the intersection of l_1 and the upper bound of u_2. (C) Upper bound for u_2 is updated to 0.8 due to the intersection of l_2 and the upper bound of u_1.

3.3.3.1 Properties of the Bound Propagation Algorithm

Let $\tilde{U}$ be the box region as the bound estimates obtained by our bound propagation algorithm. Through the following analysis, we prove that the edge screening rule by bound propagation is still safe and the algorithm terminates in a finite number of steps.

Theorem 4 *Let* $U = \{u : L \leq D^T u \leq H, -\mathbf{1} \leq u \leq \mathbf{1}\}$*, we have*

a) U *is not an empty set, which means that there is at least one solution to the inequality system of* U*.*

b) The bound propagation algorithm derives a loose bounding box for the problems (3.42)*:* $U \subseteq \tilde{U}$*.*

c) The bound propagation algorithm is guaranteed to terminate with the complexity $O(p^2)$*.*

Proof: a) If the constraint region $B(\tau, r)$ is from Theorem 2, we can see that $\theta_0^* \in B(\tau, r)$. As $G^T\theta_0^* = \lambda_0 D^T u_0$, we can get $L \leq D^T u_0 \leq H$, thus $u_0 \in U$. If we use other LASSO screening methods [10, 67, 11] to estimate the bounds of $G^T\theta^*(\lambda)$ or $D^T u^*(\lambda)$, it is easy

to derive the proof in a similar way.

b) We prove this theorem by induction. In the initial state, $\tilde{U}_0 = \{u : -1 \leq u \leq 1\}$, hence $U \subseteq \tilde{U}_0$. Assume at step t, $U \subseteq \tilde{U}_t$. We just need to prove $U \subseteq \tilde{U}_{t+1}$. The first case is that at step $t+1$, no change is made to $\tilde{U}_t$. Hence, $\tilde{U}_{t+1} = \tilde{U}_t$, and $U \subseteq \tilde{U}_{t+1}$. For the second case, if we get a tighter bound for a certain u_i, for example, $u_i - \tilde{l}_{u_i} \geq 0$ and $\tilde{l}_{u_i} > l^t_{u_i}$, where $l^t_{u_i}$ is the current bound. This tighter bound is derived from one inequality Φ in U and the bounds in U_t for non-zero u_{ϕ_i},

$$\Phi : d_{\phi_1} u_{\phi_1} + d_{\phi_2} u_{\phi_2} + ... + d_i u_i + ... + d_{\phi_Q} u_{\phi_Q} + \phi \geq 0.$$

Let $H_{u_{\phi_i}}$ represent the half space for the bound of u_{ϕ_i}, and H_Φ is the half space for the inequality Φ. With the variable elimination and replacement by bounds, we can see $H_\Phi \cap_{i=1}^{i=Q} H_{u_{\phi_i}} \subseteq H_{u_i - \tilde{l}_{u_i} \geq 0}$, where $H_{u_i - \tilde{l}_{u_i} \geq 0}$ is the half space for the bound $u_i - \tilde{l}_{u_i} \geq 0$. As $U \subseteq H_\Phi$ and $U \subseteq H_{u_{\phi_i}}, \forall i, 1 \leq i \leq Q$, we get $U \subseteq H_{u_i - \tilde{l}_{u_i}}$. We also have $U \subseteq \tilde{U}_t$ and $\tilde{U}_t \cap H_{u_i - \tilde{l}_{u_i}} = \tilde{U}_{t+1}$, therefore $U \subseteq \tilde{U}_{t+1}$. Hence, we prove that for any $t : t \geq 1$, $U \subseteq \tilde{U}_t$.

c) First we construct a regularization graph Ψ according to the inequality system U. We take each inequality (not the bounds for u_i's) in U as one vertex in Ψ. If u_i appears in two vertices, we connect the two vertices with an edge u_i.

Note that the number of inequality bounds in $\tilde{U}$ is fixed. According to Theorem 6.2 by [68], the algorithm is guaranteed to terminate since there are feasible solutions to U and we use only the bound propagation rules. In what follows, we prove that the algorithm converges in $\Omega_\Psi + 1$ iterations, where Ω_Ψ is the longest active inference path between two nodes or two edges.

If an inequality system S can improve one inequality I, we say that S implies I. Thus, it is easy to see that U implies all of the bounds in $\tilde{U}$. From Theorem 3.2 by [68], if U

implies one bound, i.e., $l_{u_i} : u_i \geq l_{u_i}$, then l_{u_i} can be obtained with a linear combination of the inequalities in U. Let U' be the subset of U that implies l_{u_i}. If we want to infer l_{u_i}, all of the bounds for the edges and nodes in the induced sub-graph corresponding U' must reach u_i in the bound propagation procedure. So there is a longest inference path in U' for the inequality l_{u_i}. As in each iteration, the bound propagation algorithm starts from a fixed node in Ψ. The traversal from any edge or node to another edge or node progresses at least one step. Therefore, it takes at most Ω_Ψ iterations to finish the path traversal, and one more iteration to finish the final update. Putting all these together, the algorithm complexity is $O(k(\Omega_\Psi + 1)p)$. Here k is the number of non-zero entries in each column of D. Since D is highly sparse, k is a small number. In the worst case, the longest inference path is p, so the algorithm complexity is $O(p^2)$.

Theorem 4 states that the bound propagation algorithm is safe for trivial L_1 item screening and can stop within $\Omega_\Psi+1$ iterations. BP has the potential to be further improved by updating only the bounds may be affected in the previous iteration. We will show that our bound propagation algorithm is effective and much more efficient than directly solving the LP problems using standard LP solvers in CPLEX [69] in our experiments.

3.3.4 Improve Screening with Transformation

Since we adopt bound propagation, and the range for θ is a sphere, the fewer variables there are in each inequality, the tighter bound we can estimate for each u_i. Hence, we can improve the accuracy and efficiency of the bound propagation algorithm by transforming D and G. Let T^1 be a transformation matrix, which satisfies

$$a)\ \tilde{D} = DT^1; \quad b)\ \tilde{G} = GT^1; \quad c)\ \theta^T\tilde{G} = \lambda u^T\tilde{D}. \tag{3.49}$$

We look for the transformation matrix T^1 so that there are fewer non-zero entries in each column of $\tilde{D}$ after transformation than in each column of the original matrix D.

Different operation matrices D will have different transformation matrices. In the following two subsections, we introduce the transformations for Generalized Fused LASSO and trend filtering problems as examples.

3.3.4.1 Transformation for Generalized Fused LASSO

For Generalized Fused LASSO, each pair of the bound inequalities in $L \leq D^T u \leq H$ corresponds to a node in the regularization graph Ψ, as shown in Figure 3.2. To find a desirable T^1, we first initialize a visiting status variable $Visit$ for each node based on its node degree. We traverse the graph Ψ starting from a node with the degree equal to one, and then we visit the adjacent nodes with the degree of two. For each visited node, we decrease its $Visit$ status by 1. We traverse along the path until the visited node is a terminal or with its $Visit$ status larger or equal to three. We then restart the traversal again with a node having $Visit = 1$, until $Visit = 0$ for each node. In this traversal process, we accumulate the labels of visited nodes, and store the current accumulated node labels to a data structure labeled as the $Data$ section for each visited node. For each node i , we use $T_{\bullet i}$ to represent the corresponding column in T^1. We set the entries of $T^1_{\bullet i}$ in the $Data$ section to one and the other entries to zero.

Theorem 5 *For a forest graph ($\lceil \frac{p}{2} \rceil \leq |E| < p-1$) and a tree graph ($|E| = p-1$), the above graph traversal process takes $|E|$ steps; and $\tilde{D}$ is a diagonal matrix with n_T all-zero columns, where n_T is the number of trees in the graph. For a general graph with $|E| > p-1$, the graph traversal process takes fewer than $|E|$ steps.*

Proof: For a simple non-loopy graph with $|E| < p-1$, the graph traversal process just goes through each edge one time. For a complex graph with loops, the traversal process goes through the edges that are not in any loop. This leads to the theorem.

For edges in a loop or connecting loops, the previous transformation cannot isolate them from other edges. But we can still get tighter bounds L and H by using node ag-

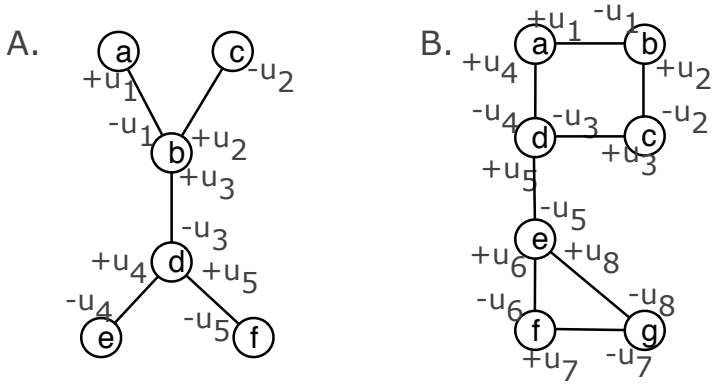


Figure 3.2: Regularization graph examples. (A) Tree graph; (B) Graph with loops. Each node corresponds to one entry in $D^T u$, with several entries in the vector u.

gregation transformation. Figure 3.2(B) illustrates one example of this kind for cyclic or loopy graphs. From Section 2.1, we can see that the bounds L and H are actually from the projection of the spherical range of θ along the direction of each $g_{\bullet i}$. For the edges in such loopy graphs, we can get better bound estimates with feature node aggregation in the given graphs. There are numerous possible combinations of feature nodes. We only consider the ones without increasing the number of variables in the resulting inequalities. One simple way is to find all the paths between the nodes with the degree higher than two, and then implement node aggregation on each path. For example, for the loopy graph in Figure 3.2(B), we can have three paths, which are $a-b-c-d$, $d-e$, and $e-f-g$. We can see that the corresponding node aggregation also corresponds to column addition in D. Hence, for such an improvement strategy, we can construct an additional transformation matrix T^2 for the graph with $|E| > p-1$.

3.3.4.2 Transformation for Trend Filtering

Similarly, we can get the transformation matrix for trend filtering. The trend filtering uses the matrix D with the structure illustrated below. Thus, we aim to reduce the number

of the involved elements in u in each row of the inequality system: $L \leq D^T u \leq H$, with a $(n-1) \times n$ matrix D:

$$D = \begin{bmatrix} 1 & -2 & 1 & ... & 0 & 0 & 0 \\ 0 & 1 & -2 & ... & 0 & 0 & 0 \\ & & & ... & & & \\ 0 & 0 & 0 & ... & 1 & -2 & 1 \end{bmatrix}. \tag{3.50}$$

We can easily derive a transformation matrix for more efficient screening with the trend filter:

$$T = \begin{bmatrix} 1 & 0 & 0 & ... & 0 & 0 & 0 \\ 2 & 1 & 0 & ... & 0 & 0 & 0 \\ 3 & 2 & 1 & ... & 0 & 0 & 0 \\ & & & ... & & & \\ n & n-1 & n-2 & ... & 3 & 2 & 1 \end{bmatrix}. \tag{3.51}$$

With this transformation, $\tilde{D}$ will become a diagonal matrix, leading to efficient screening.

3.3.5 Algorithm Flow for Sequential GL Screening and Dynamic Screening

3.3.5.1 Algorithm Flow for Sequential Screening

Given a data set $\{X, \mathbf{y}\}$, an arbitrary operation matrix D depending on the corresponding GL problem, and a sequence of decreasing λ's, our safe GL screening algorithm first derives the transformation matrix T, then applies bound propagation to iteratively tighten the bound estimates at a given λ. For each λ, features are aggregated based on the optimal solution $\beta^*(\lambda')$ of the regularized problem with the previous heavier penalty parameter λ'. Figure 3.3 illustrates the sequential screening procedure. Although for general cases of GL problems, it is difficult to compute λ_{max} to initiate the sequential screening, in the following, we list a few special cases, for which we can derive λ_{max} and $\beta^*(\lambda_{max})$ for safe

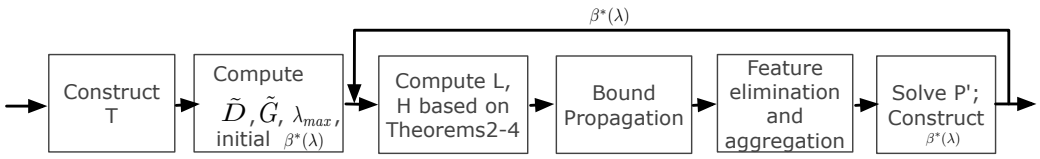


Figure 3.3: Algorithm flow for sequential GL screening.

sequential screening.

Theorem 6 *If the operation matrix D can be transformed to a diagonal matrix, i.e. $\tilde{D} = DT$, then $\lambda_{max} = \max_i |\frac{\tilde{g}_{\bullet i}^T \mathbf{f}'(G\bar{\beta})}{\tilde{d}_{ti}}|$, and $\tilde{g}_{\bullet i}$ is the ith column of $\tilde{G} = GT$, $\bar{\beta} \in \{\beta : |D\beta| = 0\}$, and $\tilde{d}_{ti}$ is the nonzero entry in the ith column of $\tilde{D}$.*

Proof: From $\tilde{G}^T\theta = \lambda\tilde{D}^T u$, we can see that if $\tilde{D}^T$is a diagonal matrix, $\lambda_{max} = \max_i |\frac{\tilde{g}_{\bullet i}^T \mathbf{f}'(G\bar{\beta})}{\tilde{d}_{ti}}|$, and $\tilde{d}_{ti}$ is the nonzero entry in ith row of $\tilde{D}^T$. As $\theta = \mathbf{f}'(G\beta)$ and $|D\beta| = 0$ at λ_{max}, we can choose $\bar{\beta}$ that $|D\bar{\beta}| = 0$. Thus, $\lambda_{max} = \max_i |\frac{\tilde{g}_{\bullet i}^T \mathbf{f}'(G\bar{\beta})}{\tilde{d}_{ti}}|$.

Based on Theorem 6, we can compute the λ_{max} for the LASSO, Fused LASSO, Tree Fused LASSO and trend filtering problems, due to the special structures of their corresponding D matrices.

3.3.5.2 Dynamic Screening

For many general cases of GL problems, it is difficult to compute λ_{max}. Thus, it is hard to derive the corresponding $\beta^*(\lambda_{max})$ to initiate the sequential screening process. Here, to solve this problem, we further propose a dynamic screening method for GL problems. Dynamic screening for LASSO [38, 39, 40] does not require the solution from a heavier penalty parameter, but constructs the constraint region for the dual variable θ based on the approximate solution from the first-order gradient method and then derive the screening rules.

If $\mathbf{f}^*(\cdot)$ is a α-strongly convex function, we have

$$\forall \theta \in \Omega_\lambda, ||\theta - \theta^*_\lambda||_2^2 \leq \frac{2}{\alpha}\Bigg[\mathbf{f}^*(\theta) - \mathbf{f}^*(\theta^*_\lambda) - \langle \mathbf{f}'^*(\theta^*_\lambda), \theta - \theta^*_\lambda\rangle\Bigg]. \tag{3.52}$$

Here θ^*_λ is the optimal solution for a given D and λ, which means $\langle \mathbf{f}^{*'}(\theta^*_\lambda), \theta - \theta^*_\lambda\rangle \geq 0$. Hence, we have

$$||\theta - \theta^*_\lambda||_2^2 \leq \frac{2}{\alpha}\Bigg[\mathbf{f}^*(\theta) - \mathbf{f}^*(\theta^*_\lambda)\Bigg].$$

Let β^*_λ be the primal solution at λ. $P(\beta, \lambda)$ represents the primal value at (β, λ). By strong duality, we have $-\mathbf{f}^*(\theta^*_\lambda) = P(\beta^*_\lambda, \lambda)$. We also know that, for any $\beta \in R^{1\times p}$, we have $P(\beta^*_\lambda, \lambda) \leq P(\beta, \lambda)$. Therefore, we can prove the following theorem,

Theorem 7 *If $\mathbf{f}^*$ is α-strongly convex, then*

$$\forall \theta \in \Omega_\lambda, \beta \in R^{1\times p}, \;\; ||\theta - \theta^*_\lambda||_2^2 \leq \frac{2}{\alpha}\Bigg[P(\beta, \lambda) + \mathbf{f}^*(\theta)\Bigg]. \tag{3.53}$$

Iterative algorithms such as the alternating direction method of multipliers (ADMM) [70] iteratively update the primal variable β, which are asymptotically close to the optimal β^*_λ. With the primal and dual relationship, for β^t at each iteration step t, we can easily compute θ^t. However, θ^t may not belong to the dual feasible region since the solution during the iterations of the ADMM algorithm is an approximate solution rather than the exact solution. To construct a constrained region for θ^*_λ using Theorem 7 so that we can apply the screening rule, we need to project θ^t to Ω_λ. Assume α is a scalar as a projection parameter. If α is small enough, we can have $\alpha\theta^t \in \Omega_\lambda$. On the other hand, we also want to have α that helps to quickly approach θ^*_λ, which will lead to tighter bounds for more effective

screening. We can formulate the optimization problem to search for α:

$$\min_{\alpha\in\Phi}\sum_{i=1}^{n} f^*(\alpha\theta_i^t),\ \Phi=\{\alpha: G^T\alpha\theta^t=\lambda D^T u, ||u||_\infty\leq 1\}. \tag{3.54}$$

If the loss function in the primal problem is linear regression, we can conveniently compute α according to the following theorem.

Theorem 8 *The scaled feasible $\hat{\theta}^t$ for any θ^t that is the closest to θ^*_λ is $\hat{\theta}^t=\alpha_t\theta^t$, where* $\alpha_t=\min\left\{\max\left\{-\min_i\frac{\lambda||D_{\bullet,i}||_1}{||X^T\theta^t||_1},-\frac{\mathbf{y}^T\theta^t}{||\theta^t||_2^2}\right\},\min_i\frac{\lambda||D_{\bullet,i}||_1}{||X^T\theta^t||_1}\right\}$.

Proof: We want to compute

$$\alpha_t=\arg\min_{\alpha\in R}\left\{\frac{1}{2}||\alpha\theta^t+y||_2^2-\frac{1}{2}y^2,\ s.t.\ X^T\alpha\theta^t=\lambda D^T u, ||u||_\infty\leq 1\right\} \tag{3.55}$$

We can see that the objective function is quadratic with a scalar variable α, and the minimum is at $\alpha=-\frac{\mathbf{y}^T\theta^t}{||\theta^t||_2^2}$. Therefore, we just need to estimate the range of α and then determine the optimal α. With the constraint $\left\{X^T\alpha\theta^t=\lambda D^T u, ||u||_\infty\leq 1\right\}$, we can see that the range of α is $\left[-\min_i\frac{\lambda||D_{\bullet,i}||_1}{||x_i^T\theta^t||_1},\min_i\frac{\lambda||D_{\bullet,i}||_1}{||x_i^T\theta^t||}\right]$. Thus the optimal least-squares objective function is minimized at $\min\left\{\max\left\{-\min_i\frac{\lambda||D_{\bullet,i}||_1}{||x_i^T\theta^t||_1},-\frac{\mathbf{y}^T\theta^t}{||\theta^t||_2^2}\right\},\min_i\frac{\lambda||D_{\bullet,i}||_1}{||x_i^T\theta^t||_1}\right\}$.

For the logistic regression or other forms of loss functions that we cannot compute the closed-form solutions for α that minimize the corresponding dual objective functions, we can choose one from $\left[-\min_i\frac{\lambda||D_{\bullet,i}||_1}{||x_i^T\theta^t||_1},\min_i\frac{\lambda||D_{\bullet,i}||_1}{||x_i^T\theta^t||_1}\right]$ that has a smaller objective function value as the projection parameter α.

With a sequence of decreasing λ's, i.e., $\lambda_0>\lambda_1>...>\lambda_n$, if we can directly compute the λ_{max} as in Theorem 10, we can start the sequential screening process from λ_{max} and then do the screening and solve the problems one by one according to the λ sequence; otherwise we can start the sequential screening process directly from λ_0 using the proposed dynamic screening procedure here.

Since it is difficult to get an absolute accurate optimum for the optimization problem at λ_i, we can add a slack variable to improve the safety of the screening at λ_{i+1},

$$\sup_{u \in U} |u_k| < 1 - \epsilon \Rightarrow D_{k\bullet}\beta^* = 0 \, . \tag{3.56}$$

In our experiments, we take $\epsilon = 0.01$. An alternative absolute safe way is to derive a relatively loose but absolute safe bound with the equation (5.2.1), i.e., $L_i = \frac{1}{\lambda}(g_{\bullet i}^T\tau - r_1||g_{\bullet i}||_2 - r_r||g_{\bullet i}||_2)$, and $H_i = \frac{1}{\lambda}(g_{\bullet i}^T\tau + r_1||g_{\bullet i}||_2 + r_2||g_{\bullet i}||_2)$, where $\tau = \hat{\theta}$, $\hat{\theta}$ is the projected dual variable at λ_i and r_1 is the ball radius from Theorem 2, and r_2 is the ball radius from (5.2.1).

3.3.6 Extensions to Models with Residual Terms

Our GL screening method can be extended to general prediction models with residual terms, such as the following problem:

$$\bar{P}: \ \min_{\beta, b} \sum_{i}^{n} f(g_{i\bullet}\beta + g_{i0}b) + \lambda||D\beta||_1. \tag{3.57}$$

Here g_{i0} could be 1, e.g., in linear regression models.

Theorem 9 *A dual form of* (3.57) *is given by*

$$\min_{\theta \in \Omega_\lambda} \sum_{i=1}^{n} f^*(H_{i\bullet}\theta), \ \Omega_\lambda = \{\theta : \bar{G}^T\theta = \lambda D^T u, ||u||_\infty \leq 1\}. \tag{3.58}$$

Here $\bar{G} = H^T G$, *and* $H = \begin{bmatrix} I \\ h \end{bmatrix}$, $h = [-\frac{g_{1,0}}{g_{n,0}}, ..., -\frac{g_{n-1,0}}{g_{n,0}}]$. *The primal and dual relationship is* $f'(g_{i\bullet}^T\beta + g_{i0}b) = H_{i\bullet}\theta_i$.

Proof: Let $G' = \big[G, g_{\bullet 0}\big]$, $D' = [D\ \mathbf{0}]$, and $\beta' = \begin{bmatrix} \beta \\ b \end{bmatrix}$. The primal problem becomes

$$\min_{\beta'} \sum_{i}^{n} f(G'\beta') + \lambda ||D'\beta'||_1. \tag{3.59}$$

By Theorem 1, the dual problem of (3.59) is

$$\min_{\theta' \in \Omega_\lambda} \sum_{i=1}^{n} f^*(\theta'_i),\ \Omega_\lambda = \{\theta' : G'^T\theta' = \lambda D'^T u, ||u||_\infty \leq 1\}. \tag{3.60}$$

As $G'^T\theta' = \lambda D'^T u$, we have $g_{\bullet 0}^T\theta' = 0$, thus $\theta'_n = -\frac{g_{1,0}}{g_{n,0}}\theta'_1 - ... - \frac{g_{n-1,0}}{g_{n,0}}\theta'_{n-1}$. Let $\theta = [\theta'_1, ..., \theta'_{n-1}]^T$, $H = \begin{bmatrix} I \\ h \end{bmatrix}$, and $h = [-\frac{g_{1,0}}{g_{n,0}}, ..., -\frac{g_{n-1,0}}{g_{n,0}}]$, and we have $\theta' = H\theta$. With $\theta' = H\theta$, the dual form becomes

$$\min_{\theta \in \Omega_\lambda} \sum_{i=1}^{n} f^*(H_{i\bullet}\theta),\ \Omega_\lambda = \{\theta : G^T H\theta = \lambda D^T u, ||u||_\infty \leq 1\}. \tag{3.61}$$

Let $\bar{\mathbf{f}}(\beta, b) = \sum_{i=1}^{n} f(g_{i\bullet}\beta + g_{i0}b)$, and $\bar{\mathbf{f}}^*(\theta) = \sum_{i=1}^{n} f^*(H_{i\bullet}\theta)$. Similarly as the proof for Theorem 2, we have the following theorem.

Theorem 10 *Let θ_0^* and θ^* be the optimal solutions to the dual problem* (3.73) *at λ_0 and λ, respectively, then we have*

$$\langle \bar{\mathbf{f}}'^*(\theta_0^*) - \frac{\lambda}{\lambda_0}\bar{\mathbf{f}}'^*(\theta^*), \theta^* - \frac{\lambda}{\lambda_0}\theta_0^* \rangle \geq 0,$$

and

$$\langle \bar{\mathbf{f}}'^*(\theta_0^*) - \bar{\mathbf{f}}'^*(\theta^*), \frac{\theta^*}{\lambda} - \frac{\theta_0^*}{\lambda_0} \rangle \geq 0.$$

If $\overline{\mathbf{f}}^*$ *is* α*-strongly convex, then*

$$||\theta^* - \theta_0^*||_2^2 \leq \frac{2}{\alpha}\left[\overline{\mathbf{f}}^*(\frac{\lambda}{\lambda_0}\theta_0^*) - \overline{\mathbf{f}}^*(\theta_0^*) + (1 - \frac{\lambda}{\lambda_0})\langle \overline{\mathbf{f}}'^*(\theta_0^*), \theta_0^*\rangle\right],$$

and

$$\forall \theta \in \Omega_\lambda, \beta \in R^{1\times p},\ \ ||\theta - \theta_\lambda^*||_2^2 \leq \frac{2}{\alpha}\left[\bar{P}(\beta, \lambda) + \overline{\mathbf{f}}^*(\theta)\right]. \tag{3.62}$$

We can construct the dual variable constraint region for sequential and dynamic screening using Theorem 10, and then, apply the transformation and bound propagation for the problems with residual terms in similar ways as discussed previously. Similarly, if the operation matrix D can be transformed into a diagonal matrix, we can compute the λ_{max} by Theorem 6 based on the dual form in Theorem 9.

3.3.7 Experiments and Discussions

In Section 7.1, we first demonstrate the advantages of using our safe GL screening method with linear regression and logistic regression on synthetic data. In Section 7.2, we compare the proposed bound propagation algorithm with the CPLEX solver to demonstrate the effectiveness and efficiency of our bound propagation algorithm for safe screening. We show the results for dynamic screening in Section 7.3. Finally we present the results of our proposed safe GL screening method on two real-world biomedical data sets: We test our screening method for Generalized Fused LASSO (GFL) linear regression on an Alzheimer's disease data FDG-PET; and then we show our results for GFL and Sparse Generalized Fused LASSO (SGFL) logistic regression on a breast cancer data.

In the following subsections, we employ CVX [71] as the base GL solver and integrate CVX with the proposed methods. We evaluate the effectiveness of our screening method

according to the rejection rate, which is defined as

$$\text{Rej. Rate} = \frac{\text{Reduced feature set size}}{\text{Original feature set size}}.$$

Let β_{cvx} be the solution from CVX, and β_{scr} as the solution from screening and CVX. We define Sparse Level and Prim. Diff. as

$$\text{Sparse Level} = \frac{\#\{i : |D\beta_{cvx}|_i < \epsilon\}}{\#\text{rows of } D},$$

and

$$\text{Prim. Diff.} = |P(\beta_{cvx}) - P(\beta_{scr})|.$$

Here $\epsilon = 10^{-5}$ and Prim. Diff. measures the difference of the primal objective function values with and without screening, which provides the evaluation of the expected safe screening. Besides average values, we also provide variation values of both Rej. rate and Prim. Diff. from running multiple experiments in respective tables.

3.3.7.1 Experiments with Synthetic Data

GFL Linear Regression (GFL-LinR) We simulate the data sets with $n = 100$ samples and $p = 3,000$ features according to a linear model $\mathbf{y} = X\beta + \epsilon$, where each column of X is a vector with random entry values in the interval $[-10, 10]$, and $\epsilon \sim N(0, 1.0)$. β takes structured relationships from a randomly generated graph G, and each element of β is in $[-1, 1]$. We simulate graph structures using both general connected graphs and forest graphs. First, we generate four different data sets. Each data set corresponds to a randomly generated graph with the total number of edge densities ranging from $p - 1$ to $1.3p$. These four data sets correspond to the rows indexed by “p-1", “1.1p", “1.2p", “1.3p" in Table 3.2, respectively. We randomly choose the variables in two subgraphs in each G to be the

non-zero contributing features. The distance between the two subgraphs is chosen to be large, so that we can independently set different β values for the corresponding features in these two subgraphs. Second, for forest graphs ($\lceil \frac{p}{2} \rceil \leq |E| < p-1$), we generate two data sets with 5 and 10 trees, respectively. These two data sets correspond to the rows of "p-10" and "p-5" in Table 3.2, respectively. Entries of β in the same tree have the same value. Table 3.2 provides the running time (in seconds) when applying GFL linear regression solved by the CVX package [71] with and without our safe GL screening method. In the table, the "GFL-LinR" column provides the time used by the CVX package without screening; "Scr." denotes the time used by our GL screening process. We have tested 52 λ values ranging from 3 to 1 in the descending order in this set of experiments to implement the sequential screening.

GFL Logistic Regression (GFL-LogR)

In this subsection we test our proposed screening method on GLF Logistic Regression. As CVX takes more time to solve the logistic regression problems, we simulate the data sets with $n = 60$, $p = 1,500$ to enable the comparison with reasonable computation time. The binary label for each sample is generated based on the logistic regression model, where we let $\tilde{\mathbf{y}} = X\beta + \epsilon$, where each column of X is a vector with random entry value belongs to $[-10, 10]$, and $\epsilon \sim N(0, 1.0)$, and we set $y_i = 1$ if $\tilde{y}_i \geq c$; and $y_i = 0$ if $\tilde{y}_i < c$. We choose the c value to give balanced training data sets in our experiments. Five data sets are simulated with the graph edge number ranging from $p-1$ to $1.3p$. The corresponding graphs and regression coefficient vectors β are generated in a similar way as GFL Linear regression with general connected graphs in the previous subsection. Table 3.3 provides the results for this study with 50 λ's ranging in $[1, 3]$, in which "GFL-LogR" denotes the implementation with the CVX package.

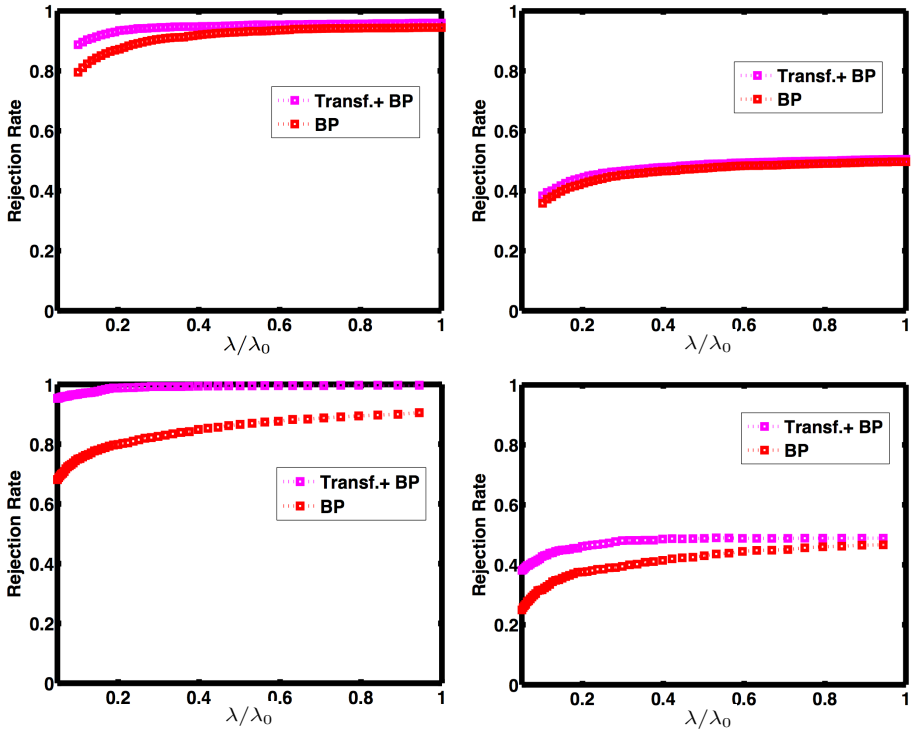


Figure 3.4: Rejection rates with and without transformation. The two plots in the first row are for GFL Linear Regression based on data from Section 7.1.1 with the edge number at $p-1$ and $1.2p$; the two plots in the second row are for GFL Logistic Regression based on data from Section 7.1.2 with the edge number at $p-1$ and $1.2p$. In the figures, "BP" stands for bound propagation, and "Transf+BP" is bound propagation with transformation.

Table 3.2: Results on synthetic data for GFL linear regression

Method	GFL-LinR (sec.)	GFL-LinR + Scr. (sec.)	Scr. (sec.)	Rej. Rate (Var.)	Sparse Level	Prim. Diff. (Var.)
p-10	950.5	160.3	38.2	0.897(4.1E-4)	[0.969, 1.000]	2.1E-7 (4.5E-15)
p-5	983.3	176.4	38.7	0.878 (9.3E-4)	[0.968, 1.000]	2.1E-8 (3.0E-16)
p-1	1549.9	137.4	33.6	0.940 (4.0E-5)	[0.967, 1.000]	1.6E-7 (1.4E-13)
1.1p	2344.5	537.2	47.7	0.699 (2.5E-5)	[0.951, 1.000]	2.2E-6 (3.6E-11)
1.2p	2284.6	924.9	55.3	0.495 (2.5E-5)	[0.960, 1.000]	4.0E-6 (4.9E-11)
1.3p	2456.0	1434.7	69.3	0.347 (1.6E-5)	[0.962, 1.000]	2.6E-6 (1.7E-11)

Table 3.3: Results on synthetic data for GFL logistic regression

Method	GFL-LogR (sec.)	GFL-LogR +Scr. (sec.)	Scr. (sec.)	Rej. Rate (Var.)	Sparse Level	Prim. Diff. (Var.)
p-1	3648.9	734.4	19.4	0.981 (2.0E-4)	[0.984, 1.000]	8.8E-9 (1.4E-14)
1.1p	4466.2	2242.7	24.8	0.611 (2.4E-3)	[0.971, 0.998]	1.5E-7 (8.4E-14)
1.2p	5167.0	3023.3	28.6	0.451 (1.3E-3)	[0.972, 1.000]	2.7E-7 (2.9E-13)
1.3p	5466.0	3423.9	28.5	0.319 (2.9E-3)	[0.971, 0.998]	4.5E-7 (3.4E-13)

Tables 3.2 and 3.3 also provide the average and variance values of Rej. Rate and Prim. Diff. with decreasing λ sequences on graphs with different edge densities. Our proposed screening method can speed up the original CVX solver up to 5 times faster. Both tables show that the screening power decreases with the graph edge density increasing. In addition, the primal objective function value difference with and without screening is negligible, indicating the safe guarantee of our screening method. For both GFL linear regression and GFL logistic regression, Figure 3.4 compares the rejection rates with and without the proposed transformation and the transformation can always improve the rejection rate to speed up solving GL problems.

Figure 3.5 gives the rejection rate for both GFL linear regression and GFL logistic regression when the graph edge number is $p-1$. In this situation, we can compute the λ_{max} according to Theorem 6. With sequential screening, many L_1 terms can be removed from both models, and the computation cost can be remarkably reduced.

SGFL Linear Regression (SGFL-LinR)

The formulation for Sparse Generalized Fused LASSO (SGFL) Linear Regression is

$$\min_{\beta} \frac{1}{2}||y - X\beta|| + \lambda_1||\beta||_1 + \lambda_2||D\beta||_1,$$

where λ_1 is the parameter that controls the sparsity penalty and λ_2 is the parameter controlling the penalty from structural feature relationships. It can be transformed into the following form:

$$\min_{\beta} \frac{1}{2}||y - X\beta|| + \lambda||\tilde{D}\beta||_1,$$

where $\tilde{D} = \begin{bmatrix} \frac{\lambda_1}{\lambda_2} I \\ D \end{bmatrix}$, and $\lambda = \lambda_2$.

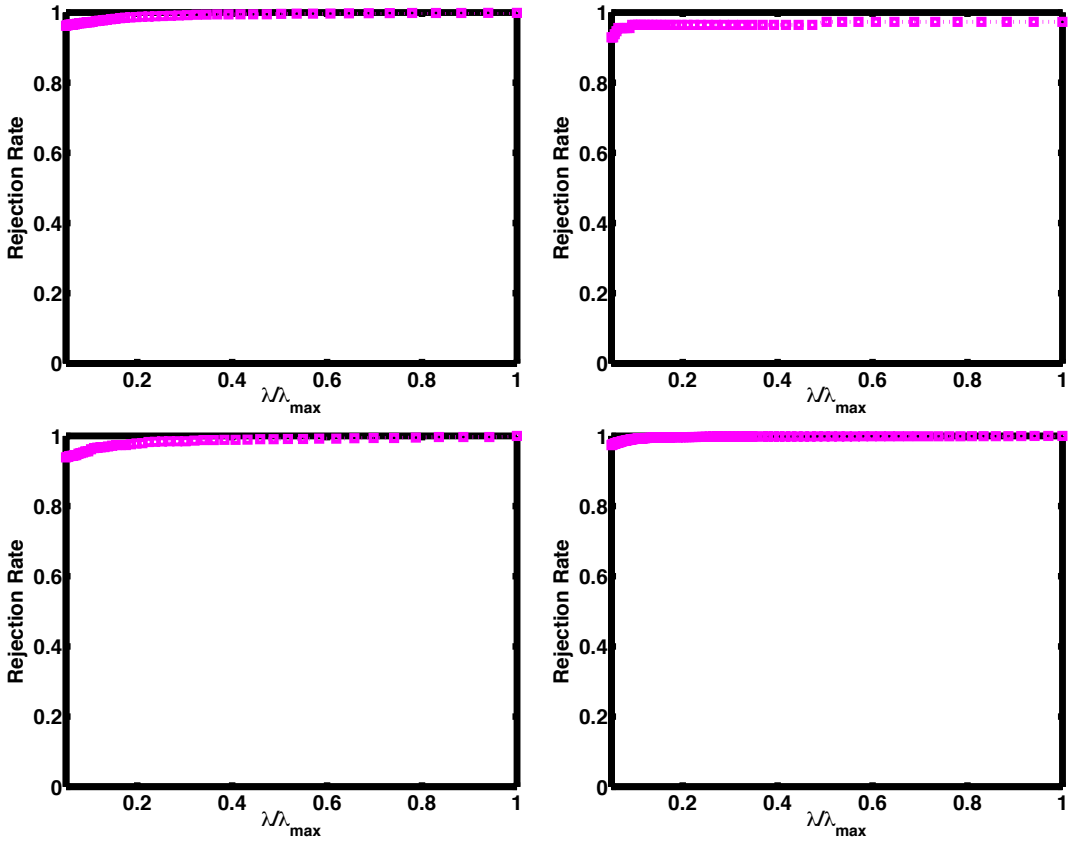


Figure 3.5: Rejection rate for GFL screening when the edge number is $p-1$. The upper left figure is for the synthetic data in Section 7.1.1; the upper right figure is for the FDG-PET data set in Section 7.4.1. The lower left figure is for the synthetic data in Section 7.1.2; and the lower right figure is for the breast cancer data in Section 7.4.2. For these four data sets, we use 50 or 100 λ's ranging from $0.05 \times \lambda_{max}$ to λ_{max}.

Table 3.4: Results on synthetic data for SGFL linear regression

Method	SGFL-LinR(sec.)	SGFL-LinR +Scr. (sec.)	Scr. (sec.)	Rej. Rate (Var.)	Sparse Level	Prim. Diff. (Var.)
0.1p	1113.8	230.0	154.6	0.962 (4.1E-5)	[0.9816, 0.9822]	4.3E-5 (1.8E-9)
0.3p	1248.1	271.0	181.8	0.949 (1.9E-5)	[0.9857, 0.9891]	1.4E-5 (6.6E-11)
0.6p	1584.3	420.8	273.1	0.892 (2.3E-4)	[0.9835, 0.9842]	4.8E-5 (3.1E-9)
0.8p	1897.8	651.4	313.3	0.861 (1.3E-4)	[0.9819, 0.9834]	6.5E-5 (3.2E-9)
p-1	2038.9	626.7	385.9	0.815 (1.4E-4)	[0.9782, 0.9805]	5.3E-5 (1.2E-9)
1.2p	2376.1	842.4	483.2	0.744 (1.2E-3)	[0.9775, 0.9793]	9.7E-5 (4.0E-9)
1.5p	3954.5	1087.0	589.0	0.681 (4.2E-4)	[0.9766, 0.9776]	9.9E-5 (5.4E-9)
1.8p	6805.5	1365.4	680.1	0.625 (6.0E-5)	[0.9781, 0.9810]	7.3E-5 (1.6E-9)
2p	9873.0	1777.2	761.7	0.567 (3.7E-4)	[0.9766, 0.9784]	8.2E-5 (1.2E-9)

We first present the results here with $\frac{\lambda_1}{\lambda_2} = 8$. To simulate the data, we generate the graphs to simulate the structural relationships among features in a similar way as in the previous two subsections and randomly set some nodes to be non-zero contributing features. We generate nine data sets with general connected graphs with the edge number ranging from $0.1p$ to $2p$. There are $n = 100$ data samples with $p = 5,000$ features in each data set. Table 3.4 gives the results for this simulation study with 51 λ's ranging from 50 to 100. "SGFL-LinR" denotes the time used by the CVX package.

Figure 3.6 further illustrates the rejection rates with changing λ_1/λ_2. We generate 11 data sets with λ_1/λ_2 changing from 0.1 to 10 for two graphs with $0.6p$ edges and $1.2p$ edges, respectively, with the other parameters fixed to the same values as described before. From the figure, we can see that the larger the difference between λ_1 and λ_2, the higher rejection rate we can get. This is expected due to the property of the inequality system in (3.38) and (3.42). For the sub-gradient u_i's with larger coefficients, they tend to have tighter bounds. If all u_i's have similar coefficients in one inequality system for their upper and lower bounds, they will have similar bound gaps as the system cannot discriminate them. In this situation, with the same system bounds, the overall screening power will be reduced.

3.3.7.2 Compare CPLEX and Bound Propagation for Safe Screening

As the goal for screening is to identify as many trivial L_1 items as possible (in other words, eliminate and/or aggregate as much as possible), we have shown that the efficacy for screening depends on the fact that how tight the bounds of the sub-gradient vector u can be estimated. Although the CPLEX LP solver can solve the bound estimation problem, it is very computationally costly. In order to clearly demonstrate that our bound propagation algorithm can achieve similar screening performance as the CPLEX LP solver with significant speedup, we compare the rejection rates as well as running time for bound estimation

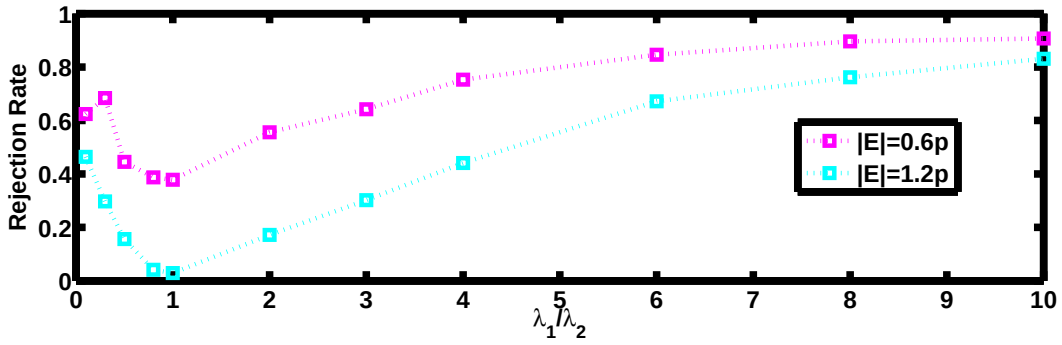


Figure 3.6: Rejection rates for SGFL Linear Regression on simulation data with different λ_1/λ_2 ratios.

based on both CPLEX LP solver and our bound propagation on a GFL-LinR model with a similarly simulated data set as in Section 7.1.1. Due to the tremendous computational cost of CPLEX, we only present the results on the data sets with $n = 50$, $p = 500$, $|E| = 1.2p$, and $|E| = 1.3p$. Table 3.5 shows the comparison of the running time. Figure 3.7 compares the rejection rates of the two methods. We can see that our bound propagation can achieve very similar rejection rates as the CPLEX LP solver, but with much lower computational cost. For bound estimation, our bound propagation algorithm can achieve speedups by two orders of magnitude compared to the CPLEX LP solver as shown in Table 3.5. In Figure 3.8, the red curves give the average values of estimated upper bound and lower bound at different λ's for bound propagation; and the blue curves are from CPLEX solver. Figure 3.9 gives the mean and variance values bound difference between CPLEX and bound propagation. From both figures, we can see that bound propagation can provide tight upper and lower bound estimates for u. We also notice that these estimates are loose bounds for u, thus they are safe for screening.

3.3.7.3 Experiments for Dynamic Screening

In this section, we take the GFL linear regression as an example to study the proposed dynamic screening. We generate 50 data samples with 1,500 features for each sample.

Table 3.5: Running time (in seconds) for CPLEX and bound propagation

Method	CPLEX	BD Propagation
CVX + GFLS (1.2p)	421.7	65.5
LP (1.2p)	369.1	2.1
CVX + GFLS (1.3p)	576.7	81.2
LP (1.3p)	508.7	2.4

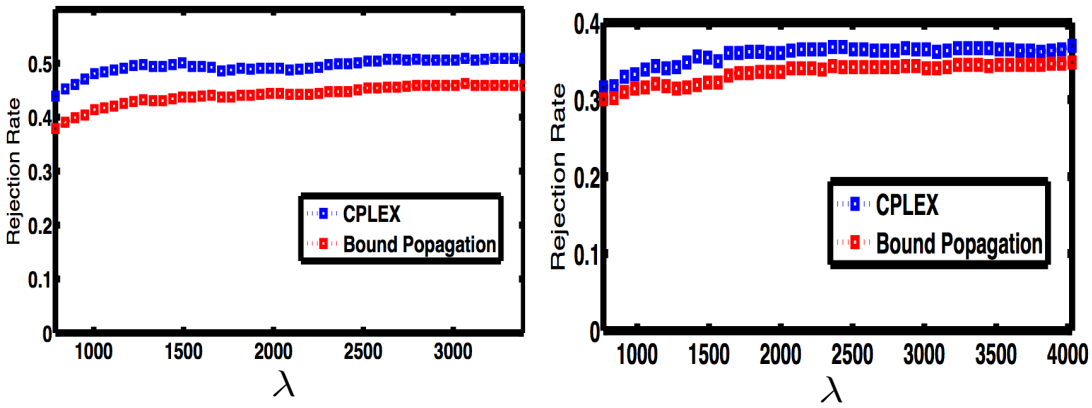


Figure 3.7: Rejection rates for CPLEX and Bound Propagation on GFL with the edge density $|E| = 1.2p$ (left) and $|E| = 1.3p$ (right) ($n = 50$ and $p = 500$).

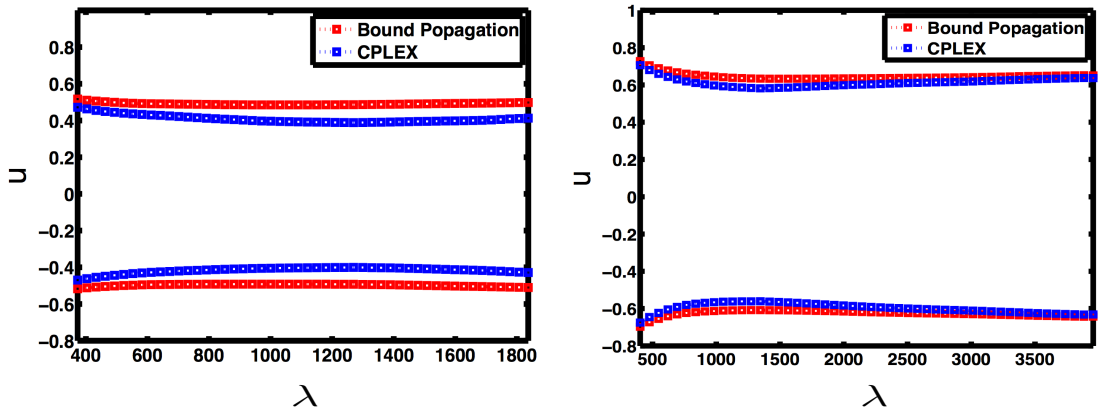


Figure 3.8: Average upper and lower bound by bound propagation and CPLEX on GFL with the edge density $|E| = 1.2p$ (left) and $|E| = 1.3p$ (right) ($n = 50$ and $p = 500$).

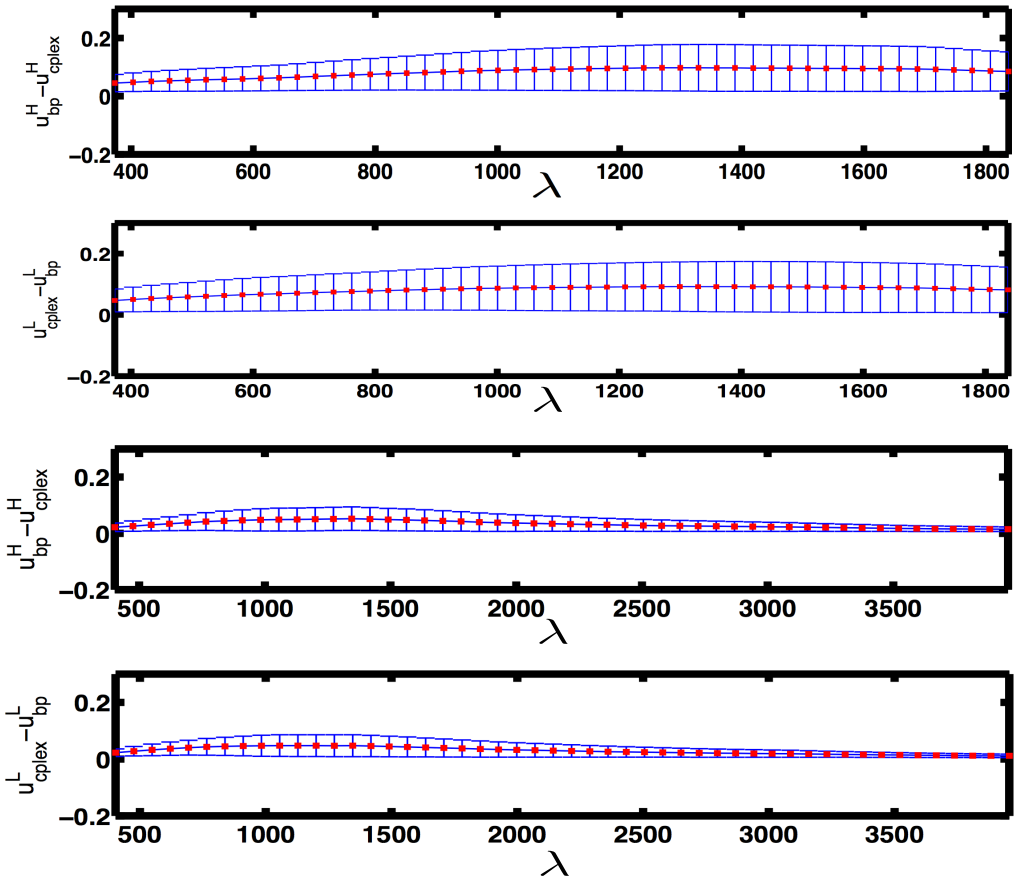


Figure 3.9: Mean and variance values for bound difference between CPLEX and bound propagation on GFL with the edge density $|E| = 1.2p$ (first two figures) and $|E| = 1.3p$ (third and forth figures) ($n = 50$ and $p = 500$).

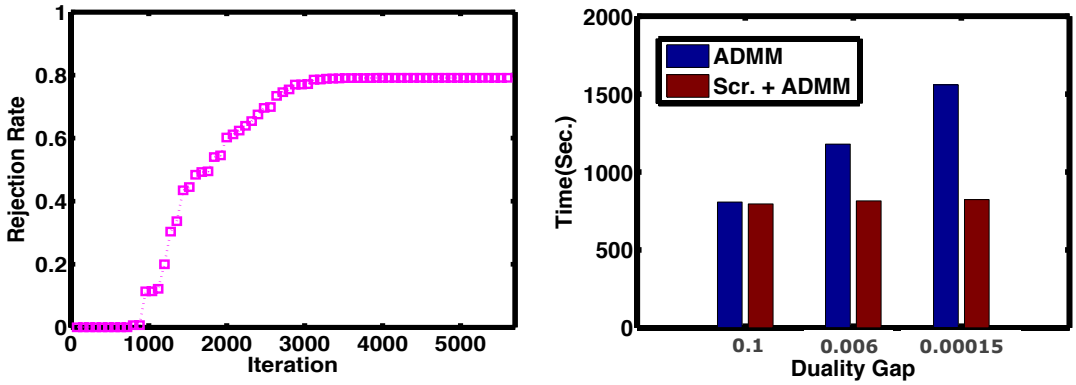


Figure 3.10: Dynamic screening. The left figure presents the number of reduced features with the increasing number of iterations. The right figure compares the running time for ADMM and ADMM with dynamic screening at different duality gap values.

We simulate the structural relationships among features by the randomly generated graph with $1.1p$ edges. The random simulation setup is the same as in Section 7.1.1. We have embedded our dynamic screening with the ADMM algorithm. Figure 3.10 shows the experimental results for the dynamic screening study. From the figure, we can see that the smaller the duality gap is, the tighter constraint region of the dual variables will be; and the more L_1 items can be removed, the more efficiency gain we can get from dynamic screening.

3.3.7.4 Experiments on Biomedical Data

We further test our GL screening method on two real-world data sets: FDG-PET [72] and Breast Cancer [54]. The first two subsections present the results for Generalized Fused LASSO on both data sets, and the last subsection gives the results for Sparse Generalized Fused LASSO (SGFL) on the breast cancer data set.

GFL Linear Regression on FDG-PET The FDG-PET data set was collected from 74 Alzheimer's disease (AD) patients, 172 mild cognitive impairment (MCI) subjects, and 81 normal control (NC) subjects, which was downloaded from the Alzheimer's Disease Neu-

Table 3.6: Results on FDG-PET data set

Method	p-1	1.1p	1.2p	1.3p
GFL-LinR	64.4	66.7	64.2	69.1
GFL-LinR + Scr.	23.1	27.1	38.2	43.0
Rej. Rate (Var.)	0.939 (6.7E-4)	0.720 (3.6E-4)	0.549 (3.6E-4)	0.394 (2.0E-4)
Sparse Level	[0.955, 1.000]	[0.967, 0.992]	[0.963, 0.993]	[0.959, 0.993]
Prim. Diff. (Var.)	3.2E-7 (7.3E-14)	2.4E-7 (2.6E-14)	2.4E-7 (3.0E-13)	4.2E-7 (2.5E-12)

roimaging Initiative (ADNI) database [73]. After preprocessing of the data by following the approach adopted in [72], 116 features (each feature corresponds to a brain region) can be derived for each subject. The outcome variable in this data set takes transformed numerical values from the original categorical sample label (NC, MCI, and AD). We further use the method described in [72] to construct the regularization graph by using the Sparse Inverse Covariance Estimation (SICE) [74]. Table 4.4 gives the running time of CVX and CVX + Screening for different scenarios, where each scenario has a different graph density controlled by SICE. Results in Table 4.4 clearly show the significant improvement on computational time if our GFL screening is applied to remove many of the trivial edges and aggregate the corresponding variables, before the use of the CVX to solve the GFL learning problem. We apply sequential screening with 54 decreasing λ's for each graph density based on SICE. Figure 3.11 gives the rejection rates with and without transformation for graphs $|E| = 1.2p$ and $|E| = 1.3p$. We can see that with transformation, we can further reduce the problem size.

GFL Logistic Regression on Breast Cancer Breast cancer data set consists of gene expression data for 8,141 genes in 295 breast cancer tumors (78 metastatic and 217 non-metastatic) [54]. The largest connected component in the human protein-protein inter-

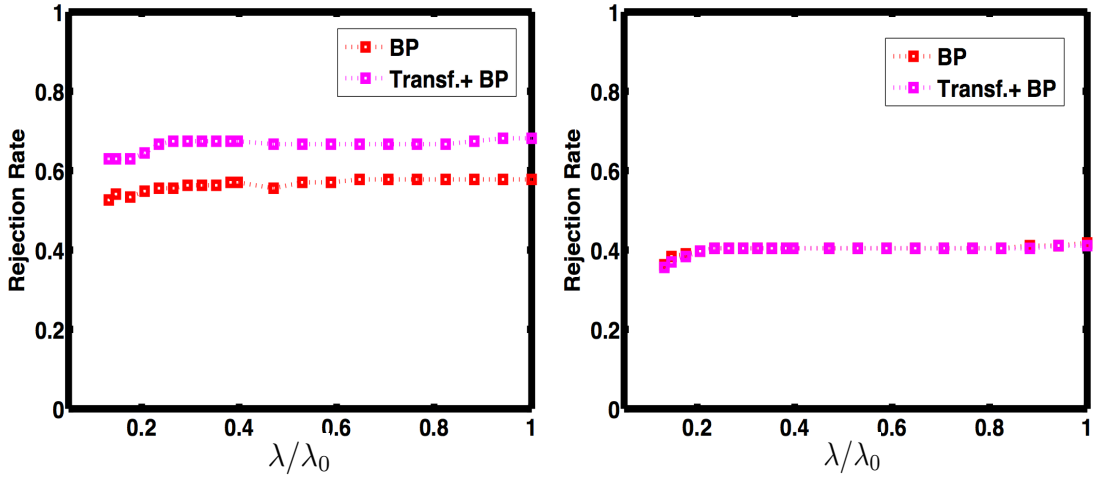


Figure 3.11: FDG-PET data with and without transformation (Left:$|E| = 1.2$, right: $|E| = 1.3$).

action (PPI) network was identified in [54] to capture the gene-gene relationships by a connected graph with 7,782 nodes. To generate different regularization graphs with different edge density levels, we start with a randomly induced tree from the PPI network and gradually add back edges randomly chosen from the original PPI network. Table 3.7 shows the running time for CVX with and without screening on these different graphs. We apply sequential screening with 64 decreasing λ's for each graph density. The bottom right plot in Figure 3.5 presents the rejection rate at 100 different λ's with edge number $p - 1$.

SGFL Logistic Regression on Breast Cancer

We also have tested the proposed screening method on the following SGFL logistic regression problem,

$$\min_{\beta} \sum_{i} \left\{ \log(1 + \exp(x_i\beta)) - x_i\beta y_i \right\} + \lambda ||\tilde{D}\beta||_1,$$

Table 3.7: Results for GFL-LogR on breast cancer data set

Method	p-1	1.1p	1.2p	1.3p
GFL-LogR	6434.1	7159.3	6849.0	6944.6
GFL-LogR+Scr.	1234.6	2504.3	3427.0	3730.9
Rej. Rate (Var.)	0.915 (5.8E-4)	0.749 (7.3E-4)	0.667 (5.3E-4)	0.601 (5.3E-4)
Sparse Level	[0.981, 0.986]	[0.981, 0.989]	[0.986, 0.992]	[0.987, 0.993]
Prim. Diff. (Var.)	1.1E-7 (4.0E-15)	5.5E-7 (2.3E-13)	5.6E-8 (2.0E-15)	7.7E-8 (5.0E-15)

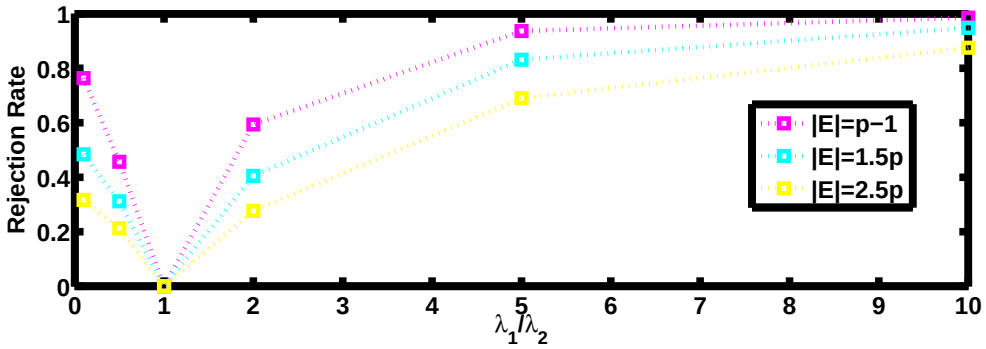


Figure 3.12: Rejection rate for SGFL Logistic Regression on breast cancer with different λ_1/λ_2.

where $\tilde{D} = \begin{bmatrix} \frac{\lambda_1}{\lambda_2} I \\ D \end{bmatrix}$, and $\lambda = \lambda_2$. This problem is the same as in Section 7.1.3 except the different loss function. We use the breast cancer data set with the same preprocessing as done in the previous subsection. Table 3.8 shows the running time for CVX with and without screening. In this study, we choose 30 λ's ranging from $0.1\lambda_0$ to λ_0, and $\lambda_0 = ||X^T\mathbf{y}||_\infty$. Figure 3.12 gives the rejection rates for different graph densities when $\frac{\lambda_1}{\lambda_2}$ is 5.

Table 3.8: Results on breast cancer data for SGFL logistic regression

Method	SGFL-Log -R(sec.)	SGFL-LogR +Scr. (sec.)	Scr. (sec.)	Rej. Rate (Var.)	Sparse Level	Prim. Diff. (Var.)
p-1	20597.0	2660.3	83.6	0.937 (2.4E-4)	[0.9997, 1.0000]	9.9E-8 (3.0E-15)
1.3p	23146.0	5085.8	110.3	0.870 (4.8E-4)	[0.9998, 1.0000]	6.0E-8 (9.5E-16)
1.7p	25787.0	7860.3	142.7	0.796 (6.8E-4)	[0.9998, 1.0000]	5.2E-8 (1.4E-15)
2p	33444.0	11995.0	168.6	0.753 (6.3E-4)	[0.9999, 1.0000]	4.3E-8 (8.4E-16)
2.5p	58271.0	26710.0	215.2	0.690 (5.8E-4)	[0.9999, 1.0000]	4.0E-8 (9.6E-16)

From the simulation and real-world data studies, we can see our screening method can effectively remove the trivial L_1 items especially when the operation matrix D can be transformed into a diagonal matrix. For SGFL problems, according to our experiments, when the difference between λ_1 and λ_2 is larger, we can obtain higher screening power, which has been similarly observed in Section 7.1.3.

The proposed methods are appealing in solving GL problems with high feature dimension, and with the D matrix diagonalizable (the number of non-zero entries in D can be reduced with column transformations), as demonstrated in the reported experiments. For example, for generalized Gused LASSO (GFL) problems with a graph capturing feature relationships, we can derive a transformation matrix T according to the method detailed in Section 4.1. Other GL problems with diagonalizable D with a transformation matrix T can potentially have high rejection rates using our proposed algorithm. We have tried our method on trend filtering problems with G being an identity matrix, and the problem can be transformed into LASSO problems with the transformation method in section 4.2. Empirically our screening method does not show significant improvement for such trend filtering problems compared to learning without screening. In fact, the CVX solver is quite efficient on solving the GL problems with the family of objective functions: $\frac{1}{2}||\mathbf{y}-\beta||_2^2+\lambda||D\beta||_1$. We have tried to integrate the proposed screening method with other GL solvers, e.g., [34, 61, 62, 57], but none of these methods can provide sufficient scalability or accuracy for screening. More details about comparison between these solvers can be found in Appendix of the supplemental file.

3.3.7.5 Comparison between CVX and Other GL Solvers

We have made a great effort to integrate the proposed screening method with other GL solvers. Methods in [34, 60, 61, 62] provide novel approaches for solving the path solution problem for GL. The screening method proposed in this chapter requires accurate solutions

Table 3.9: Compare CVX and other solvers on data sets with $p = 500, n = 30$.

Method	GraphCut [57]	Path Sol. [34]	CVX
Time(Sec.) ($\|E\| = 0.8p$)	58.7	65.9	1.2
$P(\hat{\beta})$ ($\|E\| = 0.8p$)	15.8794	14.8662	14.8630
Time(Sec.) ($\|E\| = 1.3p$)	118.9	134.2	2.4
$P(\hat{\beta})$ ($\|E\| = 1.3p$)	2.9707	2.4427	2.4246

at given λ's. In our experiments, the method in [61] cannot give primal solutions with high precision at given λ's, and cannot scale well to the problems with high dimensional feature sets, e.g. $p > 1000$. The authors in [57] also derived a solver to sparse generalized Fused lasso problems, to minimize the objective function: $\frac{1}{2}||\mathbf{y} - X\beta||_2^2 + \lambda_1||\beta||_1 + \lambda_2 \sum_{(i,j)\in E} |\beta_i - \beta_j|_1$. However, in the provided software package of [57], the penalty weight on $||\beta||_1$ can only be one constant value (λ_1). And this makes it difficult to integrate screening into the solver in [57]: During the screening process, the weights for vector $||\beta||_1$ may have different values. Furthermore, according to our experiments, similar to [61], the solver in [57] does not scale as well with the high dimensional data sets as CVX does. Table 3.9 provides the running time and primal objective values for different GL solvers on Sparse Generalized Fused LASSO (SGFL) linear regression with similarly simulated data sets as in the experiments reported in Section 7.1 when $p = 500, n = 30$. Based on these results, we have chosen the CVX solver as the baseline solution to be integrated with our screening method.

3.4 SAIF for Fused LASSO

In this section, we focus on GL problem with tree structures, which means there is no loop in the graph. We show that this type of tree Fused LASSO can be transformed into a typical LASSO form with residual term, thus we can employ the SAIF idea for scaling up.

3.4.1 Methodology

The formulation for fused LASSO is

$$\min_{\beta} \sum_{i}^{n} f(x_{i\bullet}\beta, y_i) + \lambda||D\beta||_1, \tag{3.63}$$

where $||D\beta||_1 = \sum_{(a,b)\in E} |\beta_a - \beta_b|$, and each pair in E denotes an edge in a complete tree with $\mathcal{F}$ as the vertex set. The tree $G(\mathcal{F}, E)$ captures the dependency structures among features. Here D is a matrix representation of the tree, and in each row of D, we have zeros entries except two with 1 and -1. The fused LASSO problem can be further transformed into the equivalent LASSO formulation with the following theorem.

Theorem 11 *If D can be transformed into a diagonal matrix with a column transformation matrix T, i.e. $\tilde{D} = DT$, and $\tilde{D}$ is a diagonal matrix, then*

a) the problem (3.63) *is equivalent to*

$$\tilde{P}: \ \min_{\tilde{\beta},b} \sum_{i}^{n} f\Big(\sum_{j=1}^{p-1} \tilde{x}_{ij}\tilde{\beta}_j + \tilde{x}_{ip}b, y_i\Big) + \lambda||\tilde{\beta}||_1, \tag{3.64}$$

where $\tilde{X} = XT$, and the solution relationship is $\beta^* = T\left[\begin{smallmatrix}\tilde{\beta}^* \\ b^*\end{smallmatrix}\right]$;

b) a dual form of (3.64) *is*

$$\tilde{D}: \ \min_{\bar{\theta}\in\Omega} -\sum_{i=1}^{n} f^*(-\lambda\bar{\theta}_i), \ \Omega = \big\{\bar{\theta} : |\bar{x}_i^T\bar{\theta}| \le 1, \forall i \in \{1, ..., p-1\}\big\}. \tag{3.65}$$

Here $\bar{X} = \tilde{X}_{-p}$, and $H = \begin{bmatrix} I \\ h \end{bmatrix}$, $h = \big[-\frac{\bar{x}_{1,p}}{\bar{x}_{n,p}}, ..., -\frac{\bar{x}_{n-1,p}}{\bar{x}_{n,p}}\big]$. $\bar{\theta} = H\theta_{-p}$, and $\theta = -\frac{\mathbf{f}'\left(\tilde{X}\left[\begin{smallmatrix}\tilde{\beta}^* \\ b^*\end{smallmatrix}\right]\right)}{\lambda}$. M_{-p} means matrix or vector M without pth column or entry;

c) $\lambda_{max} = \max_{i\in\{1,...,p-1\}} \left|\bar{x}_i^T\mathbf{f}'(\tilde{X}\left[\begin{smallmatrix}\mathbf{0} \\ b\end{smallmatrix}\right])\right|$.

Proof: a) The dual form for fused LASSO is

$$D_1: \sup_{\theta} -\sum_{i=1}^{n} f^*(-\lambda\theta_i) \tag{3.66}$$

$$s.t. \quad X^T\theta = D^T u \tag{3.67}$$

$$||u||_\infty \leq 1. \tag{3.68}$$

Here the primal and dual optima relation is $\theta^* = -\frac{\mathbf{f}(X\beta^*)}{\lambda}$.

With transformation matrix T, $\tilde{X} = XT$, and $\bar{D} = DT$ is a diagonal matrix, with the elements either 1 or 0 and the last column is all-zero column. And the dual form becomes

$$D_2: \sup_{\theta} -\sum_{i=1}^{n} f^*(-\lambda\theta_i) \tag{3.69}$$

$$s.t. \quad |\tilde{x}_i^T\theta| \leq 1, \forall i, 1 \leq i \leq p-1 \tag{3.70}$$

$$\tilde{x}_p^T\theta = 0 \tag{3.71}$$

We can see the corresponding primal problem for D_2 also is

$$\tilde{P}: \min_{\tilde{\beta},b} \sum_{i}^{n} f\Big(\sum_{j=1}^{p-1} \tilde{x}_{ij}\tilde{\beta}_j + \tilde{x}_{ip}b, y_i\Big) + \lambda||\tilde{\beta}||_1. \tag{3.72}$$

where $\tilde{X} = XT$, and the solution relationship is $\beta^* = T\left[\begin{smallmatrix}\tilde{\beta}^* \\ b^*\end{smallmatrix}\right]$.

b) With (3.71), we have $\theta_n = -\frac{\bar{x}_{1,p}}{\bar{x}_{n,p}}\theta_1 - ... - \frac{\bar{x}_{n-1,p}}{\bar{x}_{n,p}}\theta_{n-1}$. Let $\bar{X} = \tilde{X}_{-p}$, and $H =$

$\begin{bmatrix} I \\ h \end{bmatrix}$, $h = \big[-\frac{\bar{x}_{1,p}}{\bar{x}_{n,p}}, ..., -\frac{\bar{x}_{n-1,p}}{\bar{x}_{n,p}} \big]$, $\bar{\theta} = H\theta_{-p}$ the dual form becomes

$$\min_{\bar{\theta} \in \Omega_\lambda} -\sum_{i=1}^{n} f^*(-\lambda \bar{\theta}_i),\ \Omega = \big\{ \bar{\theta} : |\bar{x}_i^T \bar{\theta}| \leq 1, \forall i \in \{1, ..., p-1\} \big\}. \tag{3.73}$$

As we have $\theta^* = -\frac{\mathbf{f}(X\beta^*)}{\lambda}$, and $\beta^* = T\big[\begin{smallmatrix} \tilde{\beta}^* \\ b^* \end{smallmatrix} \big]$, thus we have $\bar{\theta}^* = -\frac{\big[\mathbf{f}'\big(\tilde{X}\big[\begin{smallmatrix} \tilde{\beta}^* \\ b^* \end{smallmatrix} \big]\big)\big]_{-p}}{\lambda}$.

c) As λ_{max} is the minimum λ that $\tilde{\beta}_1^* = \tilde{\beta}_2^* = ... = \tilde{\beta}_{p-1}^* = 0$, we also have $\max_{i \in \{1,...,p-1\}} |\bar{x}_i^T \bar{\theta}|$
$= 1$, $\left| \bar{x}_i^T \frac{\big[\mathbf{f}'\big(\tilde{X}\big[\begin{smallmatrix} \tilde{\beta}^* \\ b^* \end{smallmatrix} \big]\big)\big]_{-p}}{\lambda_{max}} \right| = 1$, thus $\lambda_{max} = \max_{i \in \{1,...,p-1\}} \big| \bar{x}_i^T \mathbf{f}'\big(\tilde{X}\big[\begin{smallmatrix} 0 \\ b \end{smallmatrix} \big]\big) \big|$.

With the primal form (3.64) and dual form (3.65) in Theorem 11, we just need a transformation on the feature set to apply our method to fused LASSO problems. From the proof of Theorem 2 in [38], we can easily get $\forall \bar{\theta} \in \Omega,\ \bar{\beta} = \big[\begin{smallmatrix} \tilde{\beta} \\ b \end{smallmatrix} \big] \in R^{p \times 1},\ \ ||\bar{\theta}^* - \bar{\theta}||_2^2 \leq \frac{2}{\lambda^2} \Big[\tilde{P}(\bar{\beta}) - \tilde{D}(\bar{\theta}) \Big]$. With the duality gap, we can derive the ADD and DEL rules for fused LASSO. The following Theorem shows how to project the current dual estimation $\hat{\bar{\theta}}$ to the feasible space Ω for regression with the least square loss function.

Theorem 12 *For linear regression problems with fused LASSO regularization, the scaled feasible $\hat{\bar{\theta}}$ for any θ that is the closest to $\bar{\theta}^*$ is $\hat{\bar{\theta}} = \tau\bar{\theta}$, where $\tau = min\big\{ max\big\{ \frac{\mathbf{y}^T\bar{\theta}}{\lambda||\bar{\theta}||_2^2}, -\frac{1}{||\bar{X}^T\bar{\theta}||_\infty} \big\}, \frac{1}{||\bar{X}^T\bar{\theta}||_\infty} \big\}$.*

Proof: According to Theorem 11, the dual variable corresponding to primal variable $\big[\begin{smallmatrix} \tilde{\beta} \\ b \end{smallmatrix} \big]$ is $\bar{\theta} = \{\theta_1, ..., \theta_{p-1}\}$, $\theta = -\frac{\mathbf{f}'\big(\bar{X}\big[\begin{smallmatrix} \tilde{\beta} \\ b \end{smallmatrix} \big]\big)}{\lambda}$. While $\bar{\theta}$ may not be feasible to the dual problem of linear regression. With a projection scalar τ, we try to make $\tau\bar{\theta}$ closer to $\bar{\theta}^*$ in the feasible space:

$$\tau = \arg\min_{\tau \in R} \left\{ \frac{1}{2}||\lambda\tau\bar{\theta} - \mathbf{y}||_2^2 - \frac{1}{2}||\mathbf{y}||_2^2,\ s.t.\ \ |\bar{x}_i^T \tau\bar{\theta}| \leq 1, \forall i \in \{1, ..., p-1\} \right\}. \tag{3.74}$$

From the objective function, we can easily get $\tau = \frac{\mathbf{y}^T\bar{\theta}}{\lambda||\bar{\theta}||_2^2}$ to reach the minimum point if no constraint on τ. Therefore we need to estimate the range of τ to determine the minimum for our case. From the constraint region $\big\{|\bar{x}_i^T\tau\bar{\theta}| \leq 1, \forall i \in \{1, ..., p-1\}\big\}$, the range for τ is $\big[-\frac{1}{||\bar{X}^T\bar{\theta}||_\infty}, \frac{1}{||\bar{X}^T\bar{\theta}||_\infty}\big]$. Thus $\tau = min\big\{max\{\frac{\mathbf{y}^T\bar{\theta}}{\lambda||\bar{\theta}||_2^2}, -\frac{1}{||\bar{X}^T\bar{\theta}||_\infty}\}, \frac{1}{||\bar{X}^T\bar{\theta}||_\infty}\big\}$.

The algorithm for fused LASSO is the same as LASSO with the transformation steps. As the transformation matrix is highly sparse and only have column operations on the feature matrix X, we can replace matrix multiplication with column operations to further improve computation efficiency.

3.4.2 Results for Fused LASSO

We further present the experiments fo fused LASSO with the formulation (3.63). There are a few solvers that are suitable for tree fused LASSO problems, such as [71] and the path solution method [60]. Due to the scalability and solution accuracy issues with the path solution package, we only take [71] as the baseline for comparison in our experiments. We first compare the running time between SAIF and [71] on breast cancer regarding fused LASSO linear regression; then we compare them on the FDG-PET data set [73] with logistic regression as the loss function.

3.4.2.1 Breast Cancer Data

For the same breast cancer data set, we would like to incorporate the interaction relationships among genes to formulate the fused LASSO problems for regression analysis. The largest connected component in the human protein-protein interaction (PPI) network was identified in [54] to capture the gene-gene relationships by a connected graph with 7,782 nodes. The first plot in Figure 3.13 gives the running time for both CVX and SAIF at different λ's with duality gap 1.0E-6. The results show that SAIF can significantly reduce computation cost compared with CVX.

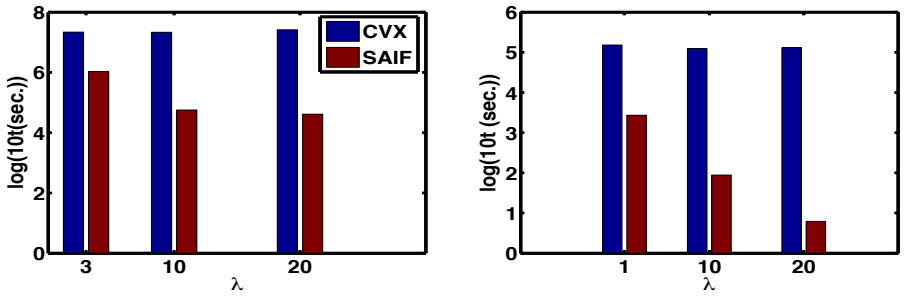


Figure 3.13: Running time for fused LASSO on breast cancer (left) and PET (right) data sets at duality gap 1.0E-6.

3.4.2.2 FDG-PET Data Set

The FDG-PET data set has 74 Alzheimer's disease (AD) patients, 172 mild cognitive impairment (MCI) subjects, and 81 normal control (NC) subjects, which was downloaded from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database [73]. 116 features (each feature corresponds to a brain region) can be derived for each subject after preprocessing. We further use the method described in [72] to construct a correlation tree on these features. We take AD as positive(1) and NC as negative(0), and disregard all of MCI samples in this set of experiments in fitting to a fused LASSO logistic regression model. The second plot in Figure 3.13 gives the running time for three λs at duality gap 10^{-6}. Again SAIF takes much less time on this data set.

3.5 Conclusions

In this chapter, we present novel safe screening methods for Generalized LASSO (GL) problems. Due to the arbitrary structure of the GL problems in terms of structural regularization, developing safe screening rules for GL problems calls for a different approach from the existing screening methods that have been devoted for special cases of the GL problems. The main idea of the first approach is to show that safe screening for GL problems can be derived by formulating equivalent dual problems constrained by linear in-

equality systems for GL learning. We then develop a novel bound propagation algorithm in the dual space to estimate tight bounds of $u^*(\lambda)$, so that we can identify as many trivial L_1 items as possible to significantly reduce the original problem size. This bound propagation method is further enhanced by novel transformation methods that can be tailored to different GL problems. The proposed propagation and transformation methods can also be applicable with dynamic screening, which further provides an efficient way to start the screening process when the desirable regularization parameter is difficult to estimate. We also show that GL problems with tree structures can be scaled up with SAIF method, in which we do not need to solve an extra problem with heavier penalty parameter. Experimental results on both synthetic and real-world data sets demonstrate the promising performance of our safe GL screening method.

4. SCALABLE ALGORITHM FOR STRUCTURED KERNEL FEATURE SELECTION *

In chapter 2 and 3, we developed two types of feature screening methods to scale up linear sparse learning. Non-linear feature selection methods have been developed to capture more complicate response relations. In this chapter, we propose one of such kind of feature selection models based on kernel methods. Incorporated with structured LASSO, the kernelized structured LASSO is an effective feature selection approach that can preserve the nonlinear input-output relationships as well as the structured sparseness. But as the data dimension increases, the method can quickly become computationally prohibitive. In this chapter we propose a stochastic optimization algorithm that can efficiently address this computational problem on account of the redundant kernel representations of the given data. Experiments on simulation data and PET 3D brain image data show that our method can achieve superior accuracy with less computational cost than existing methods.

4.1 Introduction

Feature selection has been one of the important problems to address the infamous curse of dimensionality in applying statistical learning methods to short and fat data with $n/p \ll 1$, where n and p denote the sample size and feature space dimension respectively. Penalized feature selection methods such as the Least Absolute Shrinkage and Selection Operator (LASSO) [16] provide one of effective solutions, which typically search for features that are linearly related to the output.

In order to explore potential nonlinear input-output relationships with feature selection,

*Part of this chapter is reprinted with permission from "A scalable algorithm for structured kernel feature selection" by S. Ren, S. Huang, J. Onofrey, X. Papademetris and X. Qian, 2015, 18^{th} International Conference on Artificial Intelligence and Statistics (AIStats), San Diego, CA, USA. JMLR: W&CP volume 38. Copyright 2015 by the authors.

researchers have proposed both parametric and non-parametric methods [16, 17, 32, 31]. We focus on non-parametric methods in this chapter, specifically, kernel feature selection methods. Kernel methods are arguably among the most popular tools that provide a practical way to capture nonlinear relationships. For example, Quadratic Programming Feature Selection (QPFS) [75] solves a quadratic programming problem with quadratic kernelized dependency measures. But with the increasing feature dimension, the Hessian matrix for the quadratic term may become singular and cause computational difficulty. Song *et al.* [76] proposed a greedy kernel feature selection method with forward feature selection or backward elimination strategies based on Hilbert-Smith Independent Criteria (HSIC) [77]. A related method—Hilbert-Schmidt Feature Selection (HSFS)—proposed in [29] can be considered as its continuous relaxation. HSFS was formulated as non-convex optimization problems with only local optimality guarantee from the resulting optimization algorithms. Neither the method in [76] nor HSFS can scale up with the feature dimension due to the non-convexity and complexity of their accompanying optimization problems. To address the scalability problem, Sparse Additive Models (SAM) have been proposed to efficiently solve kernel feature selection by a back-fitting algorithm [78], but it was shown that it may not perform well when features are not additively related. More recently, based on feature vector machines (FVM)[32], Yamada *et al.*[31] proposed a high-dimensional kernel feature selection method: HSIC-LASSO, in which the optimization problem can be efficiently solved by dual augmented Lagrangian(DAL) algorithm [79].

HSIC-LASSO is a feature-wise kernel method. When studying features from structured data such as images and networks for disease diagnosis, inherent structural and functional relationships among features may need to be integrated in feature selection for better accuracy, reproducibility, and interpretability. Feature-wise kernel selection methods may be further improved with better performance by considering such structural and functional relationships among features, especially when the sample size is limited. Hence, in this

chapter, we aim to develop such a kernel feature selection method that explicitly imposes structural constraints among selected features. One of such structured penalized feature selection methods is the Fused LASSO [17, 57] in linear regression and classification. The direction implementation of Fused LASSO for kernel feature selection to capture nonlinearity is computationally challenging. When the sample size and feature dimension increase, for example when studying 3-Dimensional brain images, the general batch-based optimization becomes inefficient and even infeasible. To address this computational difficulty, we introduce explicit structural constraints for structured kernel feature selection and derive a highly scalable stochastic optimization algorithm for this structured kernel feature selection method that is designed for the classification problems.

In summary, we propose a new structured kernel feature selection method based on the Hilbert-Smith Independent Criteria [77] but with explicitly enforced structural constraints to incorporate potential structural and functional relationships among features when they are available. The derived stochastic optimization algorithm is tailored to such a structured kernel feature selection problem and can efficiently solve the problem of very large size, for example for 3D brain images, on account of the redundant kernel representations of the given data. Finally, unlike HSIC-LASSO, which is designed for feature selection and requires separate learning processes for prediction with the selected features, our structured kernel feature selection method is formulated in a supervised learning framework and simultaneously learns the prediction model that can be directly adopted for new data.

The remaining of the chapter is organized as follows: Section 2 formulates the structured kernel feature selection problem; Section 3 derives the tailored stochastic optimization algorithm; Section 4 presents and discusses our experimental results with both simulation data and 3D PET brain images; Section 5 provides the discussion on the relationships of our method with the existing kernel feature selection methods in literature; Section 6 concludes this chapter and provides future research directions.

4.2 Methodology

In this section, we present our structured kernel feature selection model for classification.

4.2.1 Structured Kernel Feature Selection

Different from [31], we take the Hinge loss function in our model instead of the least squared loss in [31] since we focus on classification problems in this chapter. Without loss of generality, with the input features $X \in \mathbf{R}^{n\times p}$ and output responses $Y \in \{-1, 1\}^n$, the penalized kernel feature selection problem can be formulated as follows with the L_1-norm penalty as typically done in LASSO:

$$\min_{\mathbf{a}} \sum_{m=1}^{n}[n - \bar{L}_m^T(a_0\mathbf{1} + \sum_{i=1}^{p} a_i\bar{K}_m^i)]_+ + \lambda_1|\mathbf{a}_{1,\ldots,p}|_1 \tag{4.1}$$
$$+ \lambda_2 \sum_{(i,j)\in E} (a_i - a_j)^2$$

$$s.t. \qquad a_i \geq 0 \quad \forall i \geq 1 \tag{4.2}$$

where the first term is the Hinge Loss; $\bar{L}_m$ is a $n-$dimensional vector, corresponding to the mth column of the output kernel matrix $\tilde{L}$; and $\bar{K}_m^i$ corresponds to the mth column of $\tilde{K}^i$, which is the kernel matrix for feature $\mathbf{x}_i$. n is the number of data sample, and p is the number of feature. The structural constraints among candidate features are imposed as quadratic terms of fitting coefficients $\mathbf{a}$ in (4.1), where E denotes all the available pairwise structural relationships among features. We consider an six-neighborhood-system for 3D images. We note that these quadratic terms can be rewritten in the matrix form with the graph Laplacian based on the feature structural relationships. But for many applications, the Laplacian is highly sparse, and it is not advisable to store and use the Laplacian matrix directly in the algorithm. With the L_1-norm regularization term, the non-negative con-

straints (4.2) guarantees that the active features have larger values and non-related features have small values to make the results easily interpretable. As similarly done in [31], for each feature $\mathbf{x}_i \in X$, we have

$$\tilde{K}^i = HK^iH; \;\; H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T;$$
$$K^i_{k,\ell}(\mathbf{x}_i, \mathbf{x}_i) = exp\bigg(- \frac{(x_{ki} - x_{\ell i})^2}{2\sigma^2_{\mathbf{x}_i}}\bigg);$$
$$\bar{K}^i = vec(\tilde{K}^i); \qquad \bar{K}^i_m = \tilde{K}^i_{\bullet,m}.$$

For output responses Y, we adopt the following kernel:

$$\tilde{L} = HYY^TH; \quad \bar{L} = vec(\tilde{L}); \quad \bar{L}_m = \tilde{L}_{\bullet,m}.$$

Note that the output kernel matrix in our model is also different from the one adopted in [31], which is given as follows:

$$L(y_i, y_j) = \begin{cases} 1/n_{y_i} & \text{if } y_i = y_j \\ 0 & \text{otherwise} \end{cases}$$
$$\tilde{L} = HLH; \quad \bar{L} = vec(\tilde{L}),$$

where n_{y_i} is the number of training samples in class y_i.

4.2.2 Interpretation by Hilbert-Smith Independent Criteria

The formulated optimization problem in (4.1) aims to identify predictive features that have large inner-product values between $\bar{L}$ and $\bar{K} = a_0\mathbf{1} + \sum_i \bar{K}^i a_i$ under previously

described constraints. By expanding the inner-product $\bar{L}^T\bar{K}$, we have

$$\bar{L}^T\bar{K} = tr(\tilde{L}\tilde{K}) = a_0 tr(\tilde{L}I) + \sum_i a_i tr(\tilde{L}\tilde{K}^i) = a_0 tr(\tilde{L}I) + \sum_i a_i HSIC(Y, \mathbf{x}_i).$$

$HSIC(Y, \mathbf{x}_i) = tr(\tilde{L}\tilde{K}^i)$ is the empirical estimation of Hilbert-Smith Independent Criteria (HSIC), which is the same kernel-based independence measure adopted in HSIC [76] and HSIC-LASSO [31]. As proven in [77], $HSIC$ always takes nonnegative value and is zero if and only if the two variables are independent. When solving the optimization problem 4.1, the Hinge loss term drives the feature selection for highly correlated features with the output through the HSIC term; thereafter to have larger fitting coefficients a_i's with the nonnegative L_1-norm term penalizing less correlated or independent features to have zero coefficients. Finally, with the structural constraints, our new model can robustly recover structurally related groups of features that are responsible for the output, aiming to obtain reproducible and accurate results.

4.3 Stochastic Optimization Solution

In this section, we derive the stochastic optimization algorithm to solve our structured kernel feature selection problem.

4.3.1 Stochastic Optimization Algorithm

We note that the dimension of $\bar{K}^i$ in (4.1) is $n^2 \times 1$, and there are p such feature kernel vectors for p features in the problem. When either the sample size or feature dimension is large, many general-purpose first-order optimization algorithms cannot scale up accordingly to solve 4.1. In order to provide practical and efficient solution algorithms to 4.1, we develop a stochastic optimization algorithm based on an efficient online algorithm: the dual average method [42, 80].

As the fitting coefficients $\mathbf{a}$ are nonnegative, the optimization problem 4.1 can be

rewritten as

$$\min_{\mathbf{a}} \sum_{m=1}^{n}[n - \bar{L}_m^T(a_0\mathbf{1} + \sum_{i=1}^{p} a_i\bar{K}_m^i)]_+ + \lambda_1 \sum_{i=1}^{p} a_i \tag{4.3}$$

$$+ \lambda_2 \sum_{(i,j)\in E} (a_i - a_j)^2 \tag{4.4}$$

$$s.t. \qquad a_i \geq 0 \quad \forall i \geq 1. \tag{4.5}$$

As in the dual average method [42], the above optimization problem can be considered as two parts: the loss function part, which should be subdifferentiable; and the regularization or constraint part, which should be convex. For our current formulation 4.4, the objective function in 4.4 is subdifferentiable and can be directly taken as the loss function part for the dual average optimization. The only constraint term is the nonnegative constraints on $\mathbf{a}$. Applying the dual average method [42], the objective function can be rewritten in each step t for one sample m :

$$l_t = [n - \bar{L}_m^T(a_0\mathbf{1} + \sum_{i=1}^{p} \bar{K}_m^i a_i)]_+ + \lambda_1 \sum_{i=1}^{p} a_i + \lambda_2 \sum_{(i,j)\in E} (a_i - a_j)^2. \tag{4.6}$$

$\bar{L}_m$ and $\bar{K}_m^i$ can be considered as sample-dependent parts of $\bar{L}$ and $\bar{K}^i$, respectively.

We first compute the subgradient of l_t with respect to fitting coefficients $\mathbf{a}$:

$$\mathbf{g_t}(i) = \begin{cases} -(\bar{K}_m^i)^T\bar{L}_m + \phi(\mathbf{a}) & \text{if } \bar{L}_m^T(a_0\mathbf{1} + \sum_i \bar{K}_m^i a_i) \leq n \\ \phi(\mathbf{a}) & \text{if } \bar{L}_m^T(a_0\mathbf{1} + \sum_i \bar{K}_m^i a_i) > n \end{cases}$$

$$\phi(\mathbf{a}) = \lambda_1 + 2\lambda_2 \sum_{\{j:(i,j)\in E\}} (a_i - a_j).$$

Here, $\mathbf{g_t}(i)$ gives the ith entry of the subgradient $\mathbf{g_t}$. For a_0, $\bar{K}_m^i$ is $\mathbf{1}$. For the dual average

method at step t, we can compute the average subgradient $\bar{\mathbf{g}}_t$:

$$\bar{\mathbf{g}}_t = \frac{t-1}{t}\bar{\mathbf{g}}_{t-1} + \frac{1}{t}\mathbf{g}_t. \tag{4.7}$$

According to [42], the dual average method requires to solve a modified optimization problem by choosing a simple but strongly convex auxiliary function $h(\mathbf{a})$ as well as a nondecreasing step size sequence $\{\beta_t\}$. The appropriate choice of the auxiliary function helps make the problem smooth and strongly convex for easier optimization. The appropriate nondecreasing sequence $\{\beta_t\}$ can guarantee fast convergence. For our structured kernel feature selection problem, we need to solve the following optimization problem each step:

$$\min_{\mathbf{a}} \bar{\mathbf{g}}_t^T \mathbf{a} + \frac{\gamma(1+\ln(t))}{t}||\mathbf{a}||^2 \tag{4.8}$$

$$s.t. \quad a_i \geq 0, \forall i \geq 1. \tag{4.9}$$

Here, we take $h(\mathbf{a}) = ||\mathbf{a}||^2$ as the auxiliary function, which is strongly convex, and $\beta_t = \gamma(1+\ln(t))$. This auxiliary function $h(\mathbf{a})$ is designed specifically to have an efficient updating rule for solving our original structured kernel feature selection problem (4.1). Following the derivation of the dual average method in [42], we can prove the following theorem that gives the updating rule of our stochastic optimization algorithm.

Theorem 1 With the auxiliary function $h(\mathbf{a}) = ||\mathbf{a}||^2$ and the nondecreasing sequence $\{\beta_t\}$ with $\beta_t = \gamma(1+\ln(t))$, then the updating rule in each step t for fitting coefficients $\mathbf{a}$ for the problem (4.1) is:

$$(a_i)_t = \begin{cases} -\frac{t}{2\gamma(1+\ln(t))}\bar{\mathbf{g}}_t(i) & \text{if } i = 0 \\ [-\frac{t}{2\gamma(1+\ln(t))}\bar{\mathbf{g}}_t(i)]_+ & \text{if } i = 1, ..., p \end{cases}$$

Proof We can write the Lagrangian of the problem (4.8) by introducing the Lagrangian multipliers with the non-negative constraint:

$$L(\mathbf{a}, \lambda) = \frac{\gamma(1+\ln(t))}{t}||\mathbf{a} - (-\frac{t}{2\gamma(1+\ln(t))}\bar{\mathbf{g}}_t)||_2 - \lambda^T \mathbf{a}_{1,...,p}.$$

We can compute the gradient of the Lagrangian with respect to $\mathbf{a}$ as

$$\bigtriangledown_{\mathbf{a}} L = 2\frac{\gamma(1+\ln(t))}{t}(\mathbf{a} - (-\frac{t}{2\gamma(1+\ln(t))}\bar{\mathbf{g}}_t)) - \lambda_{1,...,p}. \tag{4.10}$$

There is no constraint for a_0. Hence, $a_0 = -\frac{t}{2\gamma(1+\ln(t))}\bar{\mathbf{g}}_t(0)$ does not violate any KKT conditions. For $a_{i:i>0}$, if $-\frac{t}{2\gamma(1+\ln(t))}\bar{\mathbf{g}}_t(i) \geq 0$, we set $a_i = -\frac{t}{2\gamma(1+\ln(t))}\bar{\mathbf{g}}_t(i)$ and $\lambda_i = 0$, and all of the KKT conditions are satisfied. If $-\frac{t}{2\gamma(1+\ln(t))}\bar{\mathbf{g}}_t(i) < 0$, we set $a_i = 0$, and $\lambda_i = \mathbf{g}_t(i)$, so $a_i\lambda_i = 0$ and also $\bigtriangledown_{\mathbf{a}} L(i) = 0$. Therefore, all of the KKT conditions can be met. With the updating rule stated in the theorem, all of the KKT conditions can be satisfied. Finally, as the problem (4.8) is convex, the updating rule in the theorem provides the optimal solution to (4.8).

This stochastic optimization algorithm provides an efficient updating rule for our original problem, and this is the key that our method can scale up to high dimensional datasets. Since the objective function in 4.1 is subdifferentiable, and the constraint set is convex, as shown in Xiao [42], with a large enough number of samples and iteration steps, the updating rules finally approach to the optimal solution to 4.1.

The pseudo-code of the final stochastic optimization algorithm is summarized in **Algorithm** 4.

Data: Data matrix X, Outcome labels Y, Feature structural relationship graph $G(V, E)$, a strongly convex auxiliary function $h(\mathbf{a})$, λ_1, λ_2.

Result: Fitting coefficients $\mathbf{a}$.

Initialization: Compute the kernel matrices for$X$ and Y; Initialize $\mathbf{a} \in \min_{\mathbf{a}} h(\mathbf{a})$;

while *Stop criteria not satisfied* **do**

1 Given the function l_t, compute the subgradient on $\mathbf{a}_t$: $\mathbf{g}_t$;

2 Update the average subgradient $\bar{\mathbf{g}}_t = \frac{t-1}{t}\bar{\mathbf{g}}_{t-1} + \frac{1}{t}\mathbf{g}_t$;

3 Calculate next $\mathbf{a}$ with

$$(a_i)_t = \begin{cases} -\frac{t}{2\gamma(1+\ln(t))}\bar{\mathbf{g}}_t(i) & \text{if } i = 0 \\ [-\frac{t}{2\gamma(1+\ln(t))}\bar{\mathbf{g}}_t(i)]_+ & \text{if } i = 1, ..., p \end{cases}$$

end

Algorithm 4: Dual Average Algorithm for Structured Kernel Feature Selection

The required storage of the kernel matrices $\tilde{K}^i$, $i = 1, ..., p$ may take large memory space for high-dimensional datasets. Similar tricks adopted in [31] can be implemented to reduce memory requirements when needed.

4.3.2 Convergence and Regret Analysis

Following [42], we can prove the following theorem:

Theorem 2 With an auxiliary function $h(\mathbf{a}) = ||\mathbf{a}||^2$, and the nondecreasing sequence $\{\beta_t\}$ with $\beta_t = \gamma(1 + \ln(t))$, Let $\{\mathbf{a}_t\}$ and $\{\mathbf{g}_t\}$ be two sequences generated by 4. Suppose the optimal solution $\mathbf{a}^*$ to problem (4.1) satisfies $h(\mathbf{a}^*) \leq D$, for some $D > 0$, and there is a constant G such that $||\mathbf{g}_t||_* \leq G$ for all $t \geq 1$, we have the following property for 4:

a) For each $t \geq 1$, the average regret is bounded by

$$R_t(\mathbf{a}) \leq \left(\gamma D^2 + \frac{G^2}{2\gamma}\right)(1 + \ln(t)).$$

b) The sequence of primal variables are bounded by

$$||\mathbf{a}_{t+1} - \mathbf{a}^*|| \leq \frac{2}{\gamma(1 + t + \ln(t))}\left(\left(\gamma D^2 + \frac{G^2}{2\gamma}\right)(1 + \ln(t)) - R_t(\mathbf{a}^*)\right).$$

Also we can have the convergence in the expectation form:

c)

$$\mathbf{E}||\mathbf{a}_{t+1} - \mathbf{a}^*|| \leq \frac{2}{1 + t + \ln(t)}\left(D^2 + \frac{G^2}{2\gamma^2}\right)(1 + \ln(t)).$$

Theorem 2(a) reveals that when $\gamma = \frac{G}{\sqrt{2}D}$, we can have the improved regret bound,

$$R_t(\mathbf{a}) = 2\sqrt{\frac{DG}{\sqrt{2}}}(1 + \ln(t)).$$

From Theorem 2(b-c), we can see that our algorithm has a convergence rate of $O(\ln(t)/t)$.

Proof: We use the indication function to represent the nonnegative region constraint:

$$\Phi(\mathbf{a}) = I_C(\mathbf{a}) = \begin{cases} 0 & \text{if } a_i \geq 0, \forall i > 0 \\ \infty & \text{if } \exists a_i < 0, i > 0 \end{cases}$$

The loss function for our original problem can be written as:

$$f(\mathbf{a}) = \sum_{m=1}^{n}[n - \bar{L}_m^T(a_0\mathbf{1} + \sum_{i=1}^{p} a_i \bar{K}_m^i)]_+ + \lambda_1 \sum_{i=1}^{p} a_i + \lambda_2 \sum_{(i,j)\in E} (a_i - a_j)^2$$

We define the region

$$\mathscr{F}_D = \{\mathbf{a} \in \text{dom}(\Phi) | h(\mathbf{a}) \leq D^2\}.$$

a) For the regret analysis, let

$$\delta_t = \max_{\mathbf{a} \in \mathscr{F}_D} \left\{ \sum_{\zeta=1}^{t} \big(\langle \mathbf{g}_\zeta, \mathbf{a}_\zeta - \mathbf{a} \rangle + \Phi(\mathbf{a}_\zeta) \big) - t\Phi(\mathbf{a})) \right\}, \;\; t = 1, 2, 3, ...$$

We can see that δ_t is the upper bound of the regret $R_t(\mathbf{a})$

$$\begin{aligned} R_t(\mathbf{a}) &= \sum_{\zeta=1}^{t} \big(f_\zeta(\mathbf{a}_\zeta) + \Phi(\mathbf{a}_\zeta) \big) - \sum_{\zeta=1}^{t} \big(f_\zeta(\mathbf{a}) + \Phi(\mathbf{a}) \big) \\ &= \sum_{\zeta=1}^{t} \big(f_\zeta(\mathbf{a}_\zeta) - f_\zeta(\mathbf{a}) + \Phi(\mathbf{a}_\zeta) \big) - t\Phi(\mathbf{a}) \\ &\leq \sum_{\zeta=1}^{t} \big(\langle \mathbf{g}_\zeta, \mathbf{a}_\zeta - \mathbf{a} \rangle + \Phi(\mathbf{a}_\zeta) \big) - t\Phi(\mathbf{a}) \\ &\leq \delta_t \end{aligned}$$

For an arbitrary initial feasible solution $\mathbf{a}_0$, we can rewrite

$$\delta_t = \sum_{\zeta=1}^{t} \big(\langle \mathbf{g}_\zeta, \mathbf{a}_\zeta - \mathbf{a}_0 \rangle + \Phi(\mathbf{a}_\zeta) \big) + \max_{\mathbf{a} \in \mathscr{F}_D} \big\{ \langle t\bar{\mathbf{g}}_t, \mathbf{a}_0 - \mathbf{a} \rangle - t\Phi(\mathbf{a}) \big\}.$$

Define $V_t(t\bar{\mathbf{g}}_t) = \max_{\mathbf{a}} \big\{ \langle t\bar{\mathbf{g}}_t, \mathbf{a} - \mathbf{a}_0 \rangle - t\Phi(\mathbf{a}) - \beta_t h(\mathbf{a}) \big\}$. As $\mathbf{a} \in \mathscr{F}_D$, we can derive the following inequality similarly as in Lemma 9 in (Xiao, 2010):

$$\delta_t \leq \sum_{\zeta=1}^{t} \big(\langle \mathbf{g}_\zeta, \mathbf{a}_\zeta - \mathbf{a}_0 \rangle + \Phi(\mathbf{a}_\zeta) \big) + V_t(-t\bar{\mathbf{g}}_t) + \beta_t D^2. \tag{4.3}$$

According to Lemmas 10 and 11 in [42], we can easily get

$$V_\zeta(-\zeta\bar{\mathbf{g}}_\zeta) + \Phi(\mathbf{a}_{\zeta+1}) \leq V_\zeta(-\zeta\bar{\mathbf{g}}_\zeta),$$

and

$$V_\zeta(-\zeta\bar{\mathbf{g}}_\zeta) \leq V_{\zeta-1}(-(\zeta-1)\bar{\mathbf{g}}_{\zeta-1}) + \langle -\mathbf{g}_\zeta, \mathbf{a}_\zeta - \mathbf{a}_0\rangle + \frac{||\mathbf{g}_\zeta||_*^2}{2(\gamma(\zeta-1)+\beta_{\zeta-1})}$$

when $\zeta \geq 2$. Hence

$$V_\zeta(-\zeta\bar{\mathbf{g}}_\zeta) + \Phi(\mathbf{a}_{\zeta+1}) \leq V_{\zeta-1}(-(\zeta-1)\bar{\mathbf{g}}_{\zeta-1}) + \langle -\mathbf{g}_\zeta, \mathbf{a}_\zeta - \mathbf{a}_0\rangle + \frac{||\mathbf{g}_\zeta||_*^2}{2(\gamma(\zeta-1)+\beta_{\zeta-1})},$$

$\zeta \geq 2$.

Moving corresponding terms, we get:

$$\langle \mathbf{g}_\zeta, \mathbf{a}_\zeta - \mathbf{a}_0\rangle + \Phi(\mathbf{a}_{\zeta+1}) \leq V_{\zeta-1}(-(\zeta-1)\bar{\mathbf{g}}_{\zeta-1}) - V_\zeta(-\zeta\bar{\mathbf{g}}_\zeta) + \frac{||\mathbf{g}_\zeta||_*^2}{2(\gamma(\zeta-1)+\beta_{\zeta-1})},$$

$\zeta \geq 2$.

When $\zeta = 1$, we have

$$\langle \mathbf{g}_1, \mathbf{a}_1 - \mathbf{a}_0\rangle + \Phi(\mathbf{a}_2) \leq - V_1(-\bar{\mathbf{g}}_1) + \frac{||\mathbf{g}_1||_*^2}{2(\beta_0)} + (\beta_0 - \beta_1)h(\mathbf{a}_2)$$

By adding all the inequalities for $\zeta = 1, ..., t$, we can get

$$\sum_{\zeta=1}^{t} \big(\langle \mathbf{g}_\zeta, \mathbf{a}_\zeta - \mathbf{a}_0\rangle + \Phi(\mathbf{a}_{\zeta+1})\big) + V_\zeta(-\zeta\bar{\mathbf{g}}_\zeta) \leq (\beta_0 - \beta_1)h(\mathbf{a}_2) + \frac{1}{2}\sum_{\zeta=1}^{t} \frac{||g_\zeta||_*^2}{\gamma(\zeta-1)+\beta_{\zeta-1}}$$

Since $\mathbf{a}_1 = \mathbf{a}_0 = \mathbf{0} \in argmin_{\mathbf{a}}\Phi(\mathbf{a})$, so $\Phi(\mathbf{a}_{t+1}) \geq \Phi(\mathbf{a}_0) = \Phi(\mathbf{a}_1)$. Adding $\Phi(\mathbf{a}_1) -$

$\Phi(\mathbf{a}_{t+1})$ to both sides,

$$\sum_{\zeta=1}^{t}\big(\langle \mathbf{g}_\zeta, \mathbf{a}_\zeta - \mathbf{a}_0\rangle + \Phi(\mathbf{a}_\zeta)\big) + V_\zeta(-\zeta\bar{\mathbf{g}}_\zeta) \leq (\beta_0-\beta_1)h(\mathbf{a}_2)+ \frac{1}{2}\sum_{\zeta=1}^{t}\frac{||g_\zeta||_*^2}{\gamma(\zeta-1)+\beta_{\zeta-1}}$$

Substituting this into (4.3) , we have

$$R_t(\mathbf{a}) \leq \delta_t \leq \beta_t D^2 + \frac{1}{2}\sum_{\zeta=1}^{t}\frac{||g_\zeta||_*^2}{\gamma(\zeta-1)+\beta_{\zeta-1}} + \frac{2(\beta_0-\beta_1)||\mathbf{g}_1||_*^2}{\beta_1+\gamma}.$$

For our algorithm $\beta_t = \gamma(1+\ln(t))$, and $\beta_0 = \beta_1 = \gamma$, hence

$$R_t(\mathbf{a}) \leq \delta_t \leq \gamma(1+\ln(t))D^2 + \frac{G^2}{2\gamma}\big(1+\sum_{\zeta=1}^{t-1}\frac{1}{\zeta+1+\ln\zeta}\big) \leq \left(\gamma D^2 + \frac{G^2}{2\gamma}\right)(1+\ln(t))$$

b) To find the bounds for primal variables, we first rewrite the solution to the subproblem (9) in the manuscript at the tth step in 4:

$$\mathbf{a}_{t+1} = \arg\min_{\mathbf{a}}\big\{\langle t\bar{\mathbf{g}}_t, \mathbf{a}\rangle + t\Phi(\mathbf{a}) + \beta_t h(\mathbf{a})\big\}.$$

The subgradients $\mathbf{b}_{t+1} \in \partial\Phi(\mathbf{a}_{t+1})$ and $\mathbf{d}_{t+1} \in \partial h(\mathbf{a}_{t+1})$ satisfy the following inequality:

$$\langle t\bar{\mathbf{g}}_t + t\mathbf{b}_{t+1} + \beta_t\mathbf{d}_{t+1}, \mathbf{a}-\mathbf{a}_{t+1}\rangle \geq 0, \forall \mathbf{a} \in \text{dom}(\Phi).$$

Since both $\Phi(\cdot)$ and $h(\cdot)$ are strongly convex, we have

$$
\begin{aligned}
&\frac{1}{2}(\gamma t+\beta_t)||\mathbf{a}_{t+1}-\mathbf{a}||^2\\
&\leq t\big(\Phi(\mathbf{a})-\Phi(\mathbf{a}_{t+1})-\langle\mathbf{b}_{t+1},\mathbf{a}-\mathbf{a}_{t+1}\rangle\big)+\beta_t\big(h(\mathbf{a})-h(\mathbf{a}_{t+1})-\langle\mathbf{d}_{t+1},\mathbf{a}-\mathbf{a}_{t+1}\rangle\big)\\
&=\beta_t h(\mathbf{a})-\beta_t h(\mathbf{a}_{t+1})-\langle t\mathbf{b}_{t+1}+\beta_t\mathbf{d}_{t+1},\mathbf{a}-\mathbf{a}_{t+1}\rangle+t\Phi(\mathbf{a})-t\Phi(\mathbf{a}_{t+1})\\
&\leq\beta_t h(\mathbf{a})-\beta_t h(\mathbf{a}_{t+1})+\langle t\bar{\mathbf{g}}_t,\mathbf{a}-\mathbf{a}_{t+1}\rangle+t\Phi(\mathbf{a})-t\Phi(\mathbf{a}_{t+1})\\
&=\beta_t h(\mathbf{a})+t\Phi(\mathbf{a})+\big\{\langle-t\bar{\mathbf{g}}_t,\mathbf{a}_{t+1}-\mathbf{a}_0\rangle-\beta_t h(\mathbf{a}_{t+1})-t\Phi(\mathbf{a}_{t+1})\big\}+\langle t\bar{\mathbf{g}}_t,\mathbf{a}-\mathbf{a}_0\rangle\\
&=\beta_t h(\mathbf{a})+t\Phi(\mathbf{a})+V_t(-t\bar{\mathbf{g}}_t)+\langle t\bar{\mathbf{g}}_t,\mathbf{a}-\mathbf{a}_0\rangle.
\end{aligned}
$$

Note that for the dual average methods in 4,

$$
\langle t\bar{\mathbf{g}}_t,\mathbf{a}-\mathbf{a}_0\rangle=\sum_{\zeta=1}^{t}\langle\mathbf{g}_\zeta,\mathbf{a}-\mathbf{a}_\zeta\rangle+\sum_{\zeta=1}^{t}\langle\mathbf{g}_\zeta,\mathbf{a}_\zeta-\mathbf{a}_0\rangle.
$$

Substituting the corresponding term, we can get

$$
\begin{aligned}
&\frac{1}{2}(\gamma t+\beta_t)||\mathbf{a}_{t+1}-\mathbf{a}||^2\\
&\leq\beta_t h(\mathbf{a})+\left\{V_t(-t\bar{\mathbf{g}}_t)+\sum_{\zeta=1}^{t}\big(\langle\mathbf{g}_\zeta,\mathbf{a}-\mathbf{a}_0\rangle+\Phi(\mathbf{a}_\zeta)\big)\right\}+\sum_{\zeta=1}^{t}\langle\mathbf{g}_\zeta,\mathbf{a}-\mathbf{a}_\zeta\rangle+t\Phi(\mathbf{a})\\
&-\sum_{\zeta=1}^{t}\Phi(\mathbf{a}_\zeta).
\end{aligned}
$$

Taking the proof for a) (4.3.2) that

$$\sum_{\zeta=1}^{t}\langle \mathbf{g}_\zeta, \mathbf{a}-\mathbf{a}_\zeta\rangle + t\Phi(\mathbf{a}) - \sum_{\zeta=1}^{t}\Phi(\mathbf{a}_\zeta) \leq \sum_{\zeta=1}^{t}\big(f_\zeta(\mathbf{a})-f_\zeta(\mathbf{a}_\zeta)\big) + t\Phi(\mathbf{a}) - \sum_{\zeta=1}^{t}\Phi(\mathbf{a}_\zeta)$$
$$= \sum_{\zeta=1}^{t}\big(f_\zeta(\mathbf{a})+\Phi(\mathbf{a})\big) - \sum_{\zeta=1}^{t}\big(f_\zeta(\mathbf{a}_\zeta)+\Phi(\mathbf{a}_\zeta)\big) = -R_t(\mathbf{a}),$$

Using (4.3.2), we can derive

$$\frac{1}{2}(\gamma t+\beta_t)||\mathbf{a}_{t+1}-\mathbf{a}||_2^2 \leq \beta_t h(\mathbf{a}) + (\beta_0-\beta_1)h(\mathbf{a}_2) + \frac{1}{2}\sum_{\zeta=1}^{t}\frac{||\mathbf{g}_\zeta||_*^2}{\gamma(\zeta-1)+\beta_{\zeta-1}} - R_t(\mathbf{a})$$

By the assumptions given in the theorem, and setting $\beta_0=\beta_1=\gamma$, we have

$$\frac{1}{2}(\gamma t+\beta_t)||\mathbf{a}_{t+1}-\mathbf{a}||_2^2 \leq \gamma(1+\ln(t))D^2 + \frac{G^2}{2\gamma}\left(1+\sum_{\zeta=1}^{t-1}\frac{1}{\zeta+1+\ln\zeta}\right) - R_t(\mathbf{a})$$
$$\leq \left(\gamma D^2+\frac{G^2}{2\gamma}\right)(1+\ln(t)) - R_t(\mathbf{a}).$$

Hence,

$$||\mathbf{a}_{t+1}-\mathbf{a}^*|| \leq \frac{2}{\gamma(1+t+\ln(t))}\left(\left(\gamma D^2+\frac{G^2}{2\gamma}\right)(1+\ln(t)) - R_t(\mathbf{a}^*)\right).$$

c) Let $z_\zeta=\{Y_\zeta, X_\zeta\}$ be the ζth sample for 4, and $\mathbf{z}[t]$ denote the collection of i.i.d random variables $\{z_1,...,z_t\}$. We can take $\mathbf{a}_\zeta$ as a function of $\{z_1,...,z_{\zeta-1}\}$, which is independent of $\{z_\zeta,...,z_t\}$.

We have

$$R_t(\mathbf{a}^*) = \sum_{\zeta=1}^{t}\big(f(\mathbf{a}_\zeta,z_\zeta)+\Phi(\mathbf{a}_\zeta)\big) - \sum_{\zeta=1}^{t}\big(f(\mathbf{a}_\zeta^*,z_\zeta)+\Phi(\mathbf{a}_\zeta^*)\big),$$

and

$$\mathbf{E}_{\mathbf{z}[t]}\big(f(\mathbf{a}_\zeta, z_\zeta) + \Phi(\mathbf{a}_\zeta)\big) = \mathbf{E}_{\mathbf{z}[\zeta-1]}\big(f(\mathbf{a}_\zeta, z_\zeta) + \Phi(\mathbf{a}_\zeta)\big) = \mathbf{E}_{\mathbf{z}[t]}\big(f(\mathbf{a}_\zeta) + \Phi(\mathbf{a}_\zeta)\big).$$

We also can get

$$\mathbf{E}_{\mathbf{z}[t]}\big(f(\mathbf{a}^*, z_\zeta) + \Phi(\mathbf{a}^*)\big) = \mathbf{E}_{z_\zeta}\big(f(\mathbf{a}^*, z_\zeta) + \Phi(\mathbf{a}^*)\big) = f(\mathbf{a}^*) + \Phi(\mathbf{a}^*).$$

Since

$$f(\mathbf{a}^*) + \Phi(\mathbf{a}^*) = \min_{\mathbf{a}} f(\mathbf{a}) + \Phi(\mathbf{a}),$$

combining the previous results leads to the following equation:

$$\mathbf{E}_{\mathbf{z}[t]} R_t(\mathbf{a}^*) = \sum_{\zeta=1}^{t} \mathbf{E}_{\mathbf{z}[t]}\big(f(\mathbf{a}_\zeta) + \Phi(\mathbf{a}_\zeta)\big) - t\big(f(\mathbf{a}^*) + \Phi(\mathbf{a}^*)\big) \geq 0.$$

Therefore, with the result from b), we can get

$$\mathbf{E}||\mathbf{a}_{t+1} - \mathbf{a}^*|| \leq \frac{2}{1 + t + \ln(t)} \left(D^2 + \frac{G^2}{2\gamma^2} \right) (1 + \ln(t)).$$

4.4 Experimental Results

We have two sets of experiments to verify the effectiveness and efficiency of out methods on structured high dimensional datasets. The first one is based on simulation experiments using MRI data. The second one is to analyze the 3D PET brain images for Alzheimer's disease (AD) prognosis [73, 57]. For these studies, we compare our algorithm with fused LASSO [17, 57], and HSIC-LASSO [31]. For fused LASSO we use the recent efficient implementation based on the graph-cut algorithm [57] with the same

efforts to provide scalable feature selection for 3D brain images.

4.4.1 Simulated Active Regions in MRI Images

In this set of experiments, we study the proposed method with a simulation of structural anomalies within MRI anatomical data. From the 1000 Functional Connectomes Project International Neuroimaging Data-Sharing Initiative[81], we randomly selected 200 3D anatomical MRI brain images from healthy subjects. Each image was spatially normalized to a $1mm \times 1mm \times 1mm$ custom, average anatomical template image using a low-dimensional free-form deformation image registration [82] with $15mm$ control point spacing. For this simulation experiments, we equally partition the total samples into healthy (negative) samples and positive samples by simulating the perturbations from the original images. Considering computation efficiency, only one brain lobe region as shown in Figure 4.1 is chosen for study. One spherical regions within the lobe are randomly perturbed as active functional areas with structural anomalies. Each voxel intensity within the active areas is modified by adding a random value g, which follows a Gaussian distribution, $N(\mu, \sigma)$. In our experiments, we take σ as the standard deviation of voxel intensity values of the original image. Among selected original images without perturbation, the average value of σ is 262.75. We perturb the voxel intensity values in 100 positive samples in a randomly selected single spherical active region with radius of $r = 4$ voxels. The images in the first row of Figure 4.1 display three axis views for one example of an original MRI image. The second and third rows in Figure 4.1 are the images after perturbation in the active areas at different levels μ.

For fused LASSO and our method we directly adopt the learned parameters for prediction as both methods are formulated as supervised learning problems. For HSIC-LASSO, kernel SVM [55] based on the learned features is used for prediction. For the proposed model, we can use the learned parameters to predict the pairwise relationship between the

test sample with all of the training samples. Since it is a binary classification problem, we can use the sign of the accumulated prediction label to determine the final prediction value. The measure on active region recovery accuracy ACC_{AR} is computed as follows:

$$ACC_{AR} = \frac{2R - ME}{2R},$$

where R denotes the number of voxels in the actual active region; and ME represents the binary voxel-wise matching error between the ground truth active region and the recovered region, which is the number of voxels in both binary images that are not in the overlap region. We take the R active voxels in the recovered region corresponding to the R voxels with highest average value f_i over all of the positive samples. When the recovered binary functional active region is the same as the ground truth region, ME = 0 and thereafter $ACC_{AR} = 1$. When the recovered region does not have any overlap voxel with the ground truth, ME = $2R$ and hence $ACC_{AR} = 0$.

In this set of experiments, 200 samples are divided into the training set and testing set. The training set contains 50 randomly chosen positive samples and 50 negative ones. The rest of the samples go to the testing set. All of the model parameters are learned based on the training set with five-fold cross validation. Since the number of training samples is not large, we use all of training samples in our stochastic algorithm without any subsampling on the training dataset. In this set of simulation experiments, we study all of the three methods on three different types of input-output relationships: linear, additive nonlinear, and non-additive nonlinear.

4.4.1.1 Linear Response

In this experiment, we compare all of the models based on simulated linear responses from perturbed MRI images with 100 positive samples having the active regions perturbed with random values following $N(\mu, \sigma)$ with $\mu = 100$, and the other 100 negative samples

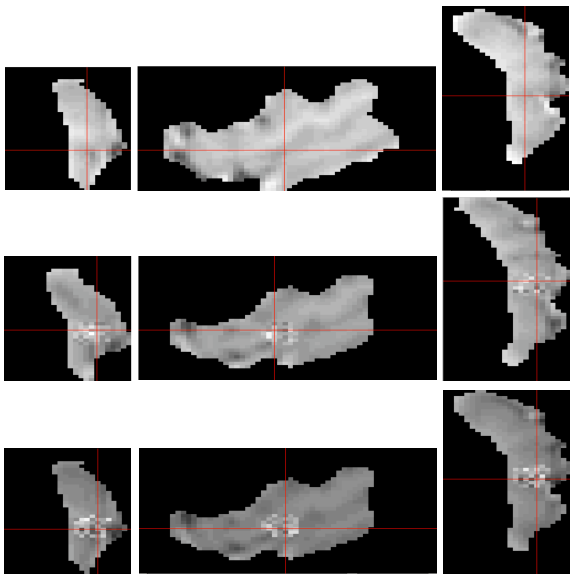


Figure 4.1: The first row shows one example from the original MRI images; the second row is the corresponding perturbed image at $\mu = 100$; the third row displays the perturbed image at $\mu = 200$.

from the original MRI images. The output label for each image is directly determined by whether there are perturbed regions. The results for the three comparing methods are shown in Table 4.1, and the recovered regions are shown in Figure 4.2.

Table 4.1: Comparison for simulated MRI images with linear responses

Method	Proposed	FL	HSIC-Lasso
Pred. Accuracy	96%	70 %	69%
Reg. Accuracy	78.1%	33.3%	23.1%
CPU time (sec.)	65.6	431.5	73.7

Table 4.1 shows that our method can achieve higher prediction accuracy as well as higher active region recovery accuracy. Moreover, our algorithm takes less computational

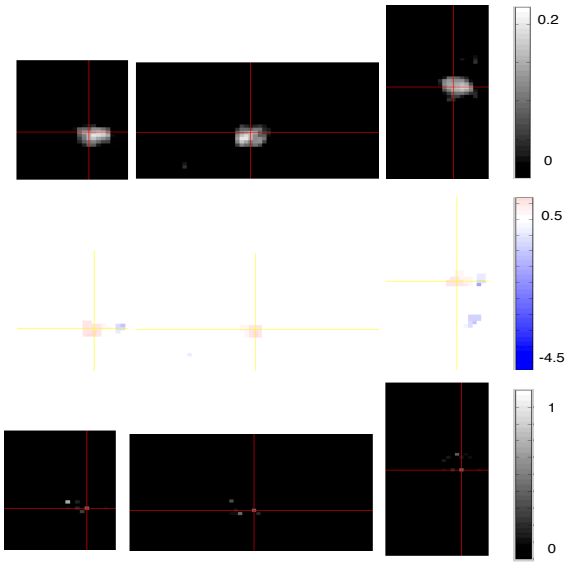


Figure 4.2: Active Regions recovered by the proposed method, Fused LASSO and HSIC-LASSO for simulated MRI images with linear responses.

resources. The results in this experiment show that our method can work robustly even though the active signal is relatively weak. The proposed model and fused LASSO can get higher ACC due to the extra structure knowledge of the data that are incorporated in the model formulation. Without the structure constraints, HSIC-LASSO misses many active voxels with the redundancy penalty term in their formulation. This is the reason why the recovered region is sparse and the ACC is low in HSIC-LASSO. We have also forced lower sparse penalty in HSIC-LASSO but it does not significantly change the results. We also note that HSIC-LASSO can achieve similar computing time compared to our proposed method due to the efficiency of their dual augmented Lagrangian (DAL) algorithm. However, HSIC-LASSO does not impose any structural constraints, which is one of bottlenecks for scalability of structured kernel feature selection.

4.4.1.2 Additive Nonlinear Response

In this experiment, we set $\mu = 200$ for perturbations. Among 200 original images, 150 are chosen to be perturbed by adding random values following $N(\mu, \sigma)$ to the corresponding voxels in the selected active regions. In addition, in order to create a nonlinear response model, not all of these samples are labelled as positive samples. We divide the voxels within the active regions into four groups: $V1, V2, V3, V4$ according to the spacial order in the image. Then we compute a nonlinear response value $\psi = \sum_{\forall v1 \in V1, v2 \in V2, v3 \in V3, v4 \in V4} sin(v1) + exp(v2/c1) + v3/c2 + (v4/c3)^2$, where $c1 = 2000, c2 = 1500$, and $c3 = 1500$ are constants in this experiment. All the perturbed images are ranked in an ascending order of ψ values. The top 100 samples are considered as positive samples while the other 100 samples are labelled as health (or negative) samples.

The results for this experiment are presented in Table 4.2. Figure 4.3 illustrates the recovered regions by three methods. It is clear that our proposed model takes lead in the accuracies and speed. The high prediction accuracy compared to the fused LASSO is due to the kernel method in our model for incorporating potential nonlinear input-output relationships. By enforcing structural constraints, our structured kernel feature selection also performs superior to HSIC-LASSO. It is interesting to note that the fused LASSO can achieve high ACC for active region recovery compared to HSIC-LASSO because of the incorporated spacial structures. However, the fused LASSO takes much longer computing time than the other two methods due to the incorporated non-smooth structure constraints even with the fast proximal and graph-cut algorithms implemented in (Xin, 2014).

Based on these simulation experiments, our structured kernel feature selection with the dual average stochastic optimization algorithm can robustly recover potential active function regions, accurately predict output responses, and scale better with both the sample

Table 4.2: Comparison for simulated MRI images with additive nonlinear responses

Method	Proposed	FL	HSIC-Lasso
Pred. Accuracy	94%	62 %	65%
Reg. Accuracy	74.5%	64.5%	27.9%
CPU time (sec.)	62.1	414.3	80.5

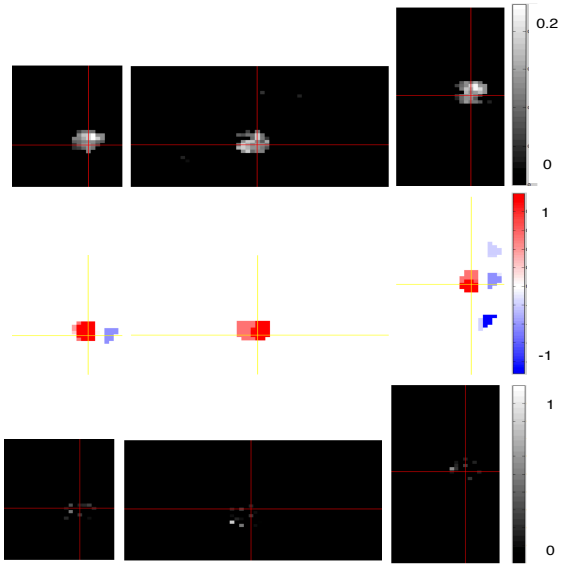


Figure 4.3: Active Regions recovered by the proposed method, Fused LASSO and HSIC-LASSO for simulated MRI images with additive nonlinear responses.

size and feature dimension compared to the other existing feature selection methods.

4.4.1.3 Non-additive Nonlinear Response

In this experiment, the simulation data is generated in a similar way as in the previous experiment. But this time we randomly choose the voxels in the four groups, and the nonlinear response value $\psi = \sum_{\forall v1 \in V1, v2 \in V2, v3 \in V3, v4 \in V4} v1 \times exp(v2/c1)/c2 + (v3/c3)^2 \times v4$, where $c1 = 2000$, $c2 = 6200$ and $c3 = 1500$. Similarly, top ranked 100 perturbed images in the ascending order of ψ are set as positive samples and the remaining 100

Table 4.3: Comparison for simulated MRI images with non-additive nonlinear responses

Method	Proposed	FL	HSIC-Lasso
Pred. Accuracy	75%	69 %	60%
Reg. Accuracy	70.9%	27.95%	0%
CPU time (sec.)	69.5	2230.4	89.9

images are negative samples.

The results of this experiment for prediction accuracies, active region recovery accuracies, and computational time are given in Table 4.3. Figure 4.4 displays the recovered regions by three methods. As visualized in the figures, our method is much more robust than the other two methods. For non-additive and nonlinear responses, the objective function is more complicated, and fused Lasso and HSIC-LASSO take longer time to reach to the optimal values. The computational time for the fused LASSO has increased dramatically. The possible reason is that as the problem becomes complicated, the line search step in the proximal algorithm in the fused LASSO takes much longer time. In this experiment, HSIC-LASSO failed to identify any responsive voxels inside the active region due to the lack of structural constraints in their formulation.

The results in this set of experiments show that our model can recover active function regions in high dimensional structured data, even when the response signal is weak and complicated.

4.4.2 PET 3D Brain Images

In this section, we test the proposed method on a 3D positron emission tomography (PET) dataset, which is collected from the Alzheimer's Disease Neuroimaging Initiative (ADNI) [73]. We collected 95 Alzheimer's disease (AD) patients and 102 healthy subjects in this set of experiments. With the affine transformation and subsequent non-linear warp-

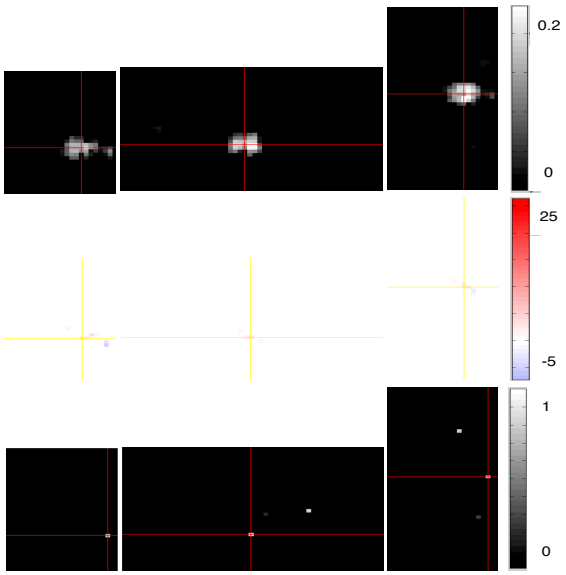


Figure 4.4: Active Regions recovered by the proposed method, Fused LASSO and HSIC-LASSO for simulated MRI images with non-additive nonlinear responses.

ing algorithm [83] in the SPM MATLAB toolbox, each image was spatially normalized to the Montreal Neurological Institute (MNI) template[84]. The data was resampled and the resolution was reduced to $4mm \times 4mm \times 4mm$ to save computation time. Student's t-test was used to remove the voxels that do not differ significantly between patients and healthy people. Furthermore, the voxels with very small intensity values are also removed to reduce computational cost. Figure 4.5 shows the mean image before and after preprocessing.

The dataset is divided into two sets: the training set contains 51 healthy people and 47 patients, the testing set has 51 healthy people and 48 patients. The parameters are learned 5 fold cross validation on the training data set according to the prediction accuracy.Table 4.4 provides the performance comparison for the three comparing methods. We can see that our method again performs much better on prediction than the other two approaches. Figure 4.6 gives the predicted active regions by three models. We use the mean of the health

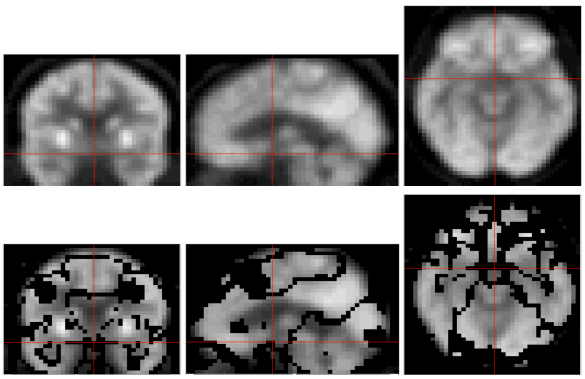


Figure 4.5: The first row displays the mean image of the original PET images in three axis views and the second row shows the corresponding mean image after preprocessing.

Table 4.4: Comparison on Pet 3D Brain Images

Method	Proposed	FL	HSIC-Lasso
Pred. Accuracy	94.9%	85.9 %	87.9%
CPU time (sec.)	163.5	2786.2	187.9

brain images as reference background, and then we add the learned voxels weights by the three models on the background. We can see our method can recover multiple regions.

4.5 Conclusions

Our structured kernel feature selection problem is specifically designed for classification with the Hinge loss function, which can be represented by HSIC terms as we show earlier. Enforcing that related features should be selected together as they have higher probability in similarly correlating the output, our structured kernel feature selection can get more robust feature selection results. In addition to the differences in formulations, we derive a tailored stochastic optimization algorithm so that the proposed method can be implemented to efficiently solve feature selection and active region recovery when we

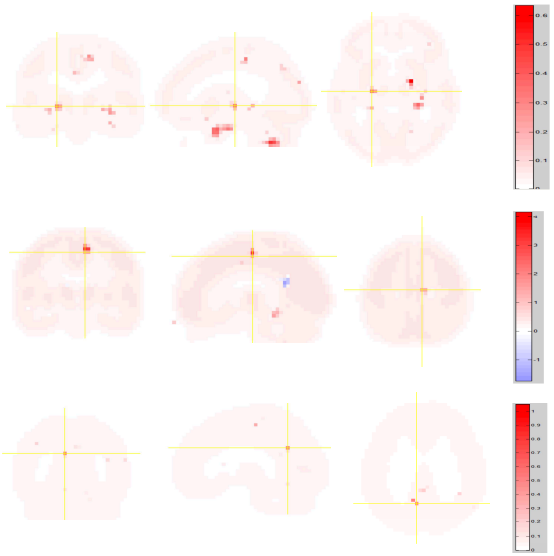


Figure 4.6: Active Regions recovered by the proposed method, Fused LASSO and HSIC-LASSO for PET 3D brain images.

have big and high-dimensional data such as 3D brain images in our experiments.

In this chapter we propose a new kernel feature selection model for binary classification problem. Based on Hilbert-Smith Independent Criteria, with the structure knowledge among features incorporated into the objective function, our model can effectively and robustly identify the active regions related to the outcome of interest. Our method can scale up to large data problem with the efficiency stochastic algorithm based on the dual average method. Experimental results on simulation data and real-world 3D image data have verified the effectiveness and efficiency of the proposed method. Our structured formulation for kernel feature selection together with the accompanying stochastic optimization method provides a practical approach for large structured data feature selection and active function region recovery from 3D brain images. Our model can be further improved with the less memory techniques [31] and faster stochastic methods [42], which will be our future research directions.

5. SCALE UP SVM WITH ACTIVE SAMPLE SELECTION

In this chapter, we propose a scalable algorithm for support vector machines (SVMs), **s**afe **a**ctive **i**ncremental support **v**ector selection (SAIV), based on the similar active incremental idea of SAIF in Chapter 2. Unlike existing working set or active set methods [24, 23, 25, 26], SAIV actively updates the active set based on the recruiting or screening rules derived from the duality gap of the sub-problem on the active set. In this way, SAIV maximally reduces the computation cost for non-support samples. Experiments on different data sets show the advantages of SAIV over the existing shrinking [24] and sequential screening methods [12].

5.1 Introduction

Similar to the sequential feature screening methods for LASSO, derived to address the prohibitive computational cost issues with extremely high-dimensional features, sequential sample screening methods for SVM have been proposed in [14, 12, 22] to address computation issues due to the extremely large number of samples. These methods estimate the range of model parameters relying on the solutions from a smaller model hyper-parameter. This type of sample screening methods have been extended to sparse SVM in [41]. Recently, the screening method developed in [15] derives sample screening rules by leveraging the duality gap, which has similar theoretical roots as the dynamic screening method for sparse learning [38].

We propose a novel method to scale up SVM to large data sets by investigating the properties of the dual problem (5.1). In (5.1), $\mathcal{D}$ is the training sample index set, and C is the model hyper-parameter as introduced in Section 1.1.2. Due to the convexity of the

SVM problem, solving the following dual problem leads to efficient SVM training:

$$D: \min_{\theta} \frac{1}{2}\theta^T Q\theta - 1^T\theta$$
$$s.t. \qquad \theta_i \in [0, C], \ \ \forall i \in \mathcal{D}.$$

Data: Training data set $\mathcal{D}$, SVM model parameter C, stopping duality gap ϵ

Result: θ

Choose l random samples from $\mathcal{D}$ as $\mathcal{A}_t$, and the rest as $\mathcal{R}_t$;

IsREC = True;

while *True* **do**

  Update θ_t regarding D_t with K iterations with $\mathcal{A}_t$ as the input ;

  Compute duality gap $G_t(\theta_t)$ based on (5.2.1) ;

  if *IsREC = False & Duality Gap* $< \epsilon$ **then**
    Stop;

  end

  SCR operation;

  if *IsREC = False* **then**
    Continue;

  else

    if $\min_{\{i \in \mathcal{A}_t} h(x_i, y_i; w_t - \sqrt{k_{ii} G_t(\theta_t)} > 0$ **then**
      IsREC = False; Continue;

    end

    REC operation;

  end

end

Put θ_t in to θ, and inflate the other entries with 0.

Algorithm 5: Active Sample Selection for SVM

Starting from a small random sample set as the active set $\mathcal{A}$, our method actively selects

and moves potential support samples (vectors) from the remaining set $\mathcal{R}$ to $\mathcal{A}$. During the iterations, non-support vectors of the sub-problem (only the samples considered in the current $\mathcal{A}$) are also removed from $\mathcal{A}$ and put into $\mathcal{R}$. With a small active set $\mathcal{A}$, CPU time and memory operations are significantly reduced compared with the existing solutions for SVM. The proposed method starts from a small active set and incrementally recruits potential support vectors. Due to its incremental nature, this approach can reduce more redundant computation compared with the existing working set and screening methods for SVM.

5.2 Safe Active Incremental Sample Selection

We first introduce two basic operations in our sample screening algorithm. We then derive our SAIV algorithms in the second sub-section.

5.2.1 REC and SCR Operations

With a feasible dual variable vector θ, the corresponding primal variable vector is $w(\theta) = Z^T\theta$. At time t, we have the active set $\mathcal{A}_t$, and the corresponding primal and dual problems for SVM with the "kernel function" ψ are P_t and $\hat{D}_t$ as follows,

$$P_t : \min_w \frac{1}{2}||w||_2^2 + C \sum_{\forall i \in \mathcal{A}_t} [1 - w^T(y_i\psi(x_i))]_+,$$

$$\hat{D}_t : \min_\theta \frac{1}{2}\theta^T Q\theta - 1^T\theta$$
$$s.t. \qquad \theta_i \in [0, C], \ \ \forall i \in \mathcal{A}_t.$$

We define the generalized primal and dual objective values as $P_t(w)$ and $\hat{D}_t(\theta)$. The dimension of w and θ can be any value not larger than n. In computing $P_t(w)$ and $\hat{D}_t(\theta)$, we inflate the missing entries in w and θ with zeros, and ignore some entries to align w

and θ with the input of P_t and $\hat{D}_t$ based on the index of original data set $\mathcal{D}$. The duality gap is defined as

$$G_t(\theta_t) = P_t(w_t) - \hat{D}_t(\theta_t) \ ,$$

where $w_t = Z_t^T \theta_t$.

Let $h(x_i, y_i; w_t) = w_t^T y_i \psi_t(x_i) - 1 = \langle Z_t^T \theta, y_i \psi_t(x_i) \rangle - 1$, and k_{ij} be the ith row and jth column entry of the kernel matrix. With $\mathcal{A}_t$ at time t, the two operations in our algorithms are defined as

REC: $\forall i \in \mathcal{R}_t$ if $h(x_i, y_i; w_t) + \sqrt{k_{ii} G_t(\theta_t)} + \sqrt{k_{jj} G_t(\theta_t)} < h(x_j, y_j; w_t), \forall j \in \mathcal{R}_t, j \neq i$, move i from $\mathcal{R}_t$ to $\mathcal{A}_t$;

SCR: $\forall i \in \mathcal{A}_t$, if $h(x_i, y_i; w_t) - \sqrt{k_{ii} G_t(\theta_t)} > 0$, move i from $\mathcal{A}_t$ to $\mathcal{R}_t$.

Our method is similar to the existing working set or active set methods. Let's use $S_{\mathcal{A}}$ to represent the set of support vector coefficients in the optimal dual solution when the working set is $\mathcal{A}$, i.e. $S = \{\theta_1^*, ..., \theta_n^*\}$. Here θ_i^* is zeros if $i \notin \mathcal{A}$. We use $\bar{\mathcal{A}}$ to represent the sample index set for the final support vectors.

Theorem 1 For active sample selection regarding the problem (5.2.1) , we have

(a) If $\min_{i \in \mathcal{R}_t} h(x_i, y_i; w_t^*) > 0$, then $S_{\mathcal{D}} = S_{\mathcal{A}_t}$.

(b) $\forall i \in \mathcal{A}_t, |h(x_i, y_i; w_t^*) - h(x_i, y_i; w_t)| \leq \sqrt{k_{ii} G_t(\theta_t)}$.

(c) For $i \in \mathcal{R}_t$, if $h(x_i, y_i; w_t) + \sqrt{k_{ii} G_t(\theta_t)} + \sqrt{k_{jj} G_t(\theta_t)} \leq h(x_j, y_j; w_t), \forall j \in \mathcal{R}_t, j \neq i$, then we have $h(x_i, y_i; w_t^*) \leq h(x_j, y_j; w_t^*), \forall j \in \mathcal{R}_t, j \neq i$.

Proof: (a) As we can see, $P(w_t^*) = P_t(w_t^*) + C \sum_{i \in \mathcal{R}_t} [1 - w_t^{*T}(y_i \psi_t(x_i))]_+ = P_t(w_t^*) + C \sum_{i \in \mathcal{R}_t} [-h(x_i, y_i; w_t^*)]_+ = P_t(w_t^*) + 0 \leq P_t(w^*) + C \sum_{i \in \mathcal{R}_t} [1 - w^{*T}(y_i \psi(x_i))]_+ = P(w^*)$, thus $w_t^* = w^*$.

(b) By applying Corollary 4.3 in [15], $\forall \{x_i, y_i\} \in \mathcal{A}_t, |h(x_i, y_i; w_t^*) - h(x_i, y_i; w_t)| = |\langle Z_t^T \theta_t^*, y_i \psi_t(x_i) \rangle - \langle Z_t^T \theta_t, y_i \psi_t(\mathbf{x}_i) \rangle| = |\langle (w_t^* - w_t),\ y_i \psi_t(x_i) \rangle| \leq ||y_i \psi_t(x_i)||_2 ||w^* -$

$w||_{\mathcal{H}} \leq \sqrt{k_{ii}G_t(\theta_t)}$.

(c) From (b), $\forall\{x_i, y_i\} \in \mathcal{R}_t$, we have $-\sqrt{k_{ii}G_t(\theta_t)} + h(x_i, y_i; w_t \leq h(x_i, y_i; w_t^*) \leq \sqrt{k_{ii}G_t(\theta_t)} + h(x_i, y_i; w_t)$. For $i, j \in \mathcal{R}_t$, if $\sqrt{k_{ii}G_t(\theta_t)} + h(x_i, y_i; w_t) \leq -\sqrt{k_{jj}G_t(\theta_t)} + h(x_j, y_j; w_t)$, which is $h(x_i, y_i; w_t) + \sqrt{k_{ii}G_t(\theta_t)} + \sqrt{k_{jj}G_t(\theta_t)} + \leq h(x_j, y_j; w_t)$, we have $h(x_i, y_i; w_t^*) \leq h(x_j, y_j; w_t^*)$.

Remark 1 Theorem 1-a) provides us a stopping criterion for the REC operation. Further more, if $\bar{\mathcal{A}} \nsubseteq \mathcal{A}_t$, then $\exists i, h(x_i, y_i; w_t^*) \leq 0$. This shows that SAIV is safe.

Data: θ_t, $G_t(\theta_t)$, $\mathcal{R}_t$, $\mathcal{A}_t$

Result: $\mathcal{R}_{t+1}$, $\mathcal{A}_{t+1}$

Set $\tilde{l} = \lceil \zeta l \rceil$;

$w_t = Z^T \theta_t$;

for $v = 1$ **to** l **do**

  $i \leftarrow \min_{i \in \mathcal{R}_t} h(x_i, y_i; w_t)$;

  Set $S_i = \{j \big| j \in \mathcal{R}_t, j \neq i, h(x_i, y_i; w_t) + \sqrt{k_{ii}G_t(\theta_t)} + \sqrt{k_{jj}G_t(\theta_t)} > h(x_j, y_j; w_t)\}$;

  if $|S_j| < \tilde{l}$ **then**

    $\mathcal{A}_t \leftarrow \mathcal{A}_t \cup \{j\}$;

    $\mathcal{R}_t \leftarrow \mathcal{R}_t - \{j\}$;

  else

    Stop;

  end

end

$\mathcal{A}_{t+1} \leftarrow \mathcal{A}_t$;

$\mathcal{R}_{t+1} \leftarrow \mathcal{R}_t$;

Algorithm 6: Algorithm for REC operation

5.2.2 Algorithm

We employ the coordinate descent method in [24] as our base iterative optimization algorithm for the dual problem $\hat{D}_t$. Algorithm 5 summaries the procedure of SAIV with the detailed steps laid out for the REC operator in Algorithm 6. Algorithms 5 and 6 are similar to the SAIF algorithm and ADD algorithm in Chapter 2, respectively.

5.3 Properties of SAIV

In this section, we first give the properties of the proposed SAIV algorithm, and then give detailed computational complexity analysis.

5.3.1 Algorithm Properties

Coordinate descent has been studied by many researchers [24, 50, 53]. The base algorithm we employed is the coordinate descent method presented in [24], in which model parameters are updated with the Gauss-Southwell Rule.

5.3.1.1 Coordinate Descent with Gauss-Southwell Rule

The following lemma gives the number of iterations needed to reach a given accuracy for the original SVM problem (5.1).

Lemma 1 With coordinate descent [24], starting from θ_0, we need at most $\log_r \frac{\varepsilon}{D(\theta_0)-D(\theta^*)}$ iterations to reach accuracy $\varepsilon = D(\theta_k) - D(\theta^*)$ for objective in (5.1). Here $r = 1 - \frac{\mu}{Ln}$, L is the coordinate wise Lipschitz continuousness value, and u is the convexity value of the loss function in (5.1).

Proof: With Gauss-Southwell Rule [85], the convergence rate is

$$\frac{D(\theta_{k+1}) - D(\theta^*)}{D(\theta_k) - D(\theta^*)} \leq r = 1 - \frac{\mu}{Ln}.$$

Staring from θ_0 to reach accuracy gap ε, we can recurrently apply (5.3.1.1),

$$\frac{D(\theta_m) - D(\theta^*)}{D(\theta_0) - D(\theta^*)} \leq r^a = \frac{\varepsilon}{D(\theta_0) - D(\theta^*)}, \implies \quad a = \log_r \frac{\varepsilon}{D(\theta_0) - D(\theta^*)}.$$

With the iteration numberlarger than a, the accuracy gap will be smaller than ε.

The duality gap converges with primal updating.

$$\begin{aligned} G(\theta_t) &= P(w(\theta_t)) - \hat{D}(\theta_t) \\ &= \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}[1 - w^T(y_i Q(x_i))]_+ + \frac{1}{2}\theta_t^T \Phi \theta_t - 1^T\theta_t \\ &= \theta_t^T \Phi \theta_t + C\sum_{i=1}^{l}[-h(x_i, y_i; w(\theta_t))]_+ - 1^T\theta_t. \end{aligned}$$

With $\lim_{k\to\infty} ||\theta_t - \theta^*|| = 0$, we have $\lim_{k\to\infty} G(\theta_k) = 0$.

5.3.1.2 Finite Numbers of REC and SCR Operations

The following theorem indicates that REC and SRC operations can end within a finite number of steps in SAIV to include all of the actual support vectors in the original SVM problem with all of training samples.

Theorem 2 Let w_t^* and θ_t^* be the optimal primal and dual solutions for the sub-problem with the active feature set $\mathcal{A}_t$.

(a) If sample $i = argmin_{i\in M_t} h(x_i, y_i; w)$ is added to $\mathcal{A}_t$ operation at time t, and $\bar{\mathcal{A}} \nsubseteq \mathcal{A}_t$, then $\theta_{t+1}^*(i) \neq 0$.

(b) If $\bar{\mathcal{A}} \nsubseteq \mathcal{A}_t$, and we have REC operation at t, then $\forall t', t < t'$, $\mathcal{A}_t \neq \mathcal{A}_{t'}$.

(c) $\exists T,\ \forall t \geq T, \theta_t^* = \theta^*$, and $w_t^* = w^*$.

Proof: (a) If $\theta_{t+1}^*(i) = 0$, then $h(x_i, y_i; w_t^*) > 0$, and with Theorem 1-a), this means $\bar{\mathcal{A}} \subseteq \mathcal{A}_t$, this contradicts with the conditions.

(b) With an REC operation at t to movie sample i into $\mathcal{A}_t$, we insert an entry into θ_t^*

corresponding to sample i with the value of 0. From a), we know that $\theta^*_{t+1}(i) \neq \tilde{\theta}^*_t(i)$.

$$D(\tilde{\theta}^*_{t+1}) = D_{t+1}(\theta^*_{t+1}) = \min_{\theta_{t+1}} D_{t+1}(\theta_{t+1}) < D_{t+1}(\begin{bmatrix} \theta^*_t \\ 0 \end{bmatrix}) = D(\tilde{\theta}^*_t)$$

SCR operation does not change dual objective value. Thus $\forall t', t' > t$, $D(\theta^*_{t'}) < D(\tilde{\theta}^*_t)$, which means $\mathcal{A}_{t'} \neq \mathcal{A}_t$.

(c) With REC operation the dual objective value always goes down. From Remark 1, there are alway samples for REC operation before working set includes $\bar{\mathcal{A}}$. Thus $\lim_{t\to\infty} \theta^*_t = \theta^*$, and $\lim_{t\to\infty} w^*_t = w^*$. With REC operations, $\mathcal{A}_t$ changes with t with finite combination according to a). Thus $\exists T,\ \forall t \geq T, \theta^*_t = \theta^*$, and $w^*_t = w^*$.

In the next section, we present detail complexity analysis for SAIV.

5.3.2 Algorithm Complexity Analysis

We consider the running time for the proposed method has three parts: sample increasing, sample screening, and accuracy pursuing, and we use T_a, T_b, and T_c to represent the corresponding time complexity, and the overall time complexity is $T = T_a + T_b + T_c$.

5.3.2.1 Sample Recruiting

To move i from $\mathcal{R}_t$ to $\mathcal{A}_t$, we need $h(x_i, y_i; w_t) + \sqrt{k_{ii}G(\theta_t)} + \sqrt{k_{jj}G(\theta_t)} < h(x_j, y_j; w_t), \forall j \in \mathcal{A}_t, j \neq i$. This leads to

$$G(\theta_t) \leq \Big(\frac{h(x_j, y_j; w_t) - h(x_i, y_i; w_t)}{\sqrt{k_{jj}} + \sqrt{k_{ii}}}\Big)^2 \approx \Big(\frac{h(x_j, y_j; w^*_t) - h(x_i, y_i; w^*_t)}{\sqrt{k_{jj}} + \sqrt{k_{ii}}}\Big)^2.$$

Samples may be added or removed from $\mathcal{A}_t$ during the sample recruiting phase. Let $\Psi_t(\theta) = D(\theta) - D(\theta^*_t)$ be the gap with REC operation at time t, we have

Lemma 2 The time complexity for the sample recruiting phase is $O\Big(\frac{2K+n}{K}\big(\frac{L}{\mu}\Phi_2 +$

$\frac{L}{\mu} n_{T_I}^2 \log \frac{\bar{\Omega}_2}{\Psi_{T_I}(\theta_{T_I})}\big) + \frac{1}{K}\big(\frac{L}{\mu}\Phi_3 + \frac{L}{\mu} n_{T_I}^3 \log \frac{\bar{\Omega}_3}{\Psi_{T_I}(\theta_{T_I})}\big)\Big)$, where

$$\Phi_2 = \sum_{t=1}^{T_I-1} n_t^2 \log \frac{\Psi_{t+1}(\theta_t)}{\Psi_t(\theta_t)},\ \bar{\Omega}_2 = \big(\Pi_{t=0}^{T_I-1} \Psi_{t+1}^{n_{t+1}^2 - n_t^2}\big)^{\frac{1}{n_{T_I}^2}},$$
$$\Phi_3 = \sum_{t=1}^{T_I-1} n_t^3 \log \frac{\Psi_{t+1}(\theta_t)}{\Psi_t(\theta_t)},\ \text{and } \bar{\Omega}_3 = \big(\Pi_{t=0}^{T_I-1} \Psi_{t+1}^{n_{t+1}^3 - n_t^3}\big)^{\frac{1}{n_{T_I}^3}}.$$

Proof: For sample recruiting phase,

$$\begin{aligned} T_a &= \sum_{t=1}^{T_I} \frac{\log_{r_t} \frac{\Psi_t(\theta_t)}{\Psi_t(\theta_{t-1})}}{K}(2Kn_t + n_t^2 + n_t n) \\ &= \frac{2K+n}{K}\sum_{t=1}^{T_I} n_t \log_{r_t} \frac{\Psi_t(\theta_t)}{\Psi_t(\theta_{t-1})} + \frac{1}{K}\sum_{t=1}^{T_I} n_t^2 \log_{r_t} \frac{\Psi_t(\theta_t)}{\Psi_t(\theta_{t-1})}. \end{aligned}$$

Let

$$\begin{aligned} T_{a1} &= \sum_{t=1}^{T_I} n_t \log_{r_t} \frac{\Psi_t(\theta_t)}{\Psi_t(\theta_{t-1})} = \sum_{t=1}^{T_I} \log_{r_t} \frac{\Psi_t^{n_t}(\theta_t)}{\Psi_t^{n_t}(\theta_{t-1})} \\ &= -\log_{r_1} \Psi_1^{n_1}(\theta_0) + \sum_{t=1}^{T_I-1} \big(\log_{r_t} \Psi_t^{n_t}(\theta_t) - \log_{r_{t+1}} \Psi_{t+1}^{n_{t+1}}(\theta_t)\big) + \log_{r_{T_I}} \Psi_{T_I}^{n_{T_I}}(\theta_{T_I}) \\ &= -\log_{r_1} \Psi_1^{n_1}(\theta_0) + \log_{r_{T_I}} \Psi_{T_I}^{n_{T_I}}(\theta_{T_I}) + \sum_{t=1}^{T_I-1} \log_{r_t} \frac{\Psi_t^{n_t}(\theta_t)}{\Psi_{t+1}^{\frac{n_{t+1}}{\log_{r_t} r_{r+1}}}(\theta_t)}. \end{aligned}$$

With

$$\log_{r_t} r_{t+1} = \frac{\log r_{t+1}}{\log r_t} = \frac{\log(1 - \frac{\mu}{n_{t+1}L})}{\log(1 - \frac{\mu}{n_t L})} \approx \frac{-\frac{\mu}{n_{t+1}L}}{-\frac{\mu}{n_t L}} = \frac{n_t}{n_{t+1}},$$

we get

$$
\begin{aligned}
T_{a1} &\approx -\log_{r_1} \Psi_1^{n_1}(\theta_0) + \log_{r_{T_I}} \Psi_{T_I}^{n_{T_I}}(\theta_{T_I}) + \sum_{t=1}^{T_I-1} \log_{r_t} \frac{\Psi_t^{n_t}(\theta_t)}{\Psi_{t+1}^{\frac{n_{t+1}^2}{n_t}}(\theta_t)} \\
&\leq -\log_{r_1} \Psi_1^{n_1}(\theta_0) + \log_{r_{T_I}} \Psi_{T_I}^{n_{T_I}}(\theta_{T_I}) + \sum_{t=1}^{T_I-1} \frac{n_t L}{\mu} \log \frac{\Psi_{t+1}^{\frac{n_{t+1}^2}{n_t}}(\theta_t)}{\Psi_t^{n_t}(\theta_t)} \\
&\leq -\log_{r_1} \Psi_1^{n_1}(\theta_0) - \frac{L}{\mu} \log \Psi_{T_I}^{n_{T_I}^2}(\theta_{T_I}) + \frac{L}{\mu} \sum_{t=1}^{T_I-1} \log \frac{\Psi_{t+1}^{n_{t+1}^2}(\theta_t)}{\Psi_t^{n_t^2}(\theta_t)} \\
&= -\log_{r_1} \Psi_1^{n_1}(\theta_0) + \frac{L}{\mu} \sum_{t=1}^{T_I-1} n_t^2 \log \frac{\Psi_{t+1}(\theta_t)}{\Psi_t(\theta_t)} + \frac{L}{\mu} \log \frac{\Pi_{t=1}^{T_I-1} \Psi_{t+1}^{n_{t+1}^2 - n_t^2}}{\Psi_{T_I}^{n_{T_I}^2}(\theta_{T_I})} \\
&= \frac{L}{\mu} \sum_{t=1}^{T_I-1} n_t^2 \log \frac{\Psi_{t+1}(\theta_t)}{\Psi_t(\theta_t)} + \frac{L}{\mu} \log \frac{\Pi_{t=0}^{T_I-1} \Psi_{t+1}^{n_{t+1}^2 - n_t^2}}{\Psi_{T_I}^{n_{T_I}^2}(\theta_{T_I})} - \log_{r_1} \Psi_1^{n_1}(\theta_0) - \frac{L}{\mu} \log \Psi_1^{n_1^2}(\theta_0).
\end{aligned}
$$

Let

$$
\begin{aligned}
\Phi_2 &= \sum_{t=1}^{T_I-1} n_t^2 \log \frac{\Psi_{t+1}(\theta_t)}{\Psi_t(\theta_t)}, \quad \bar{\Omega}_2 = \left(\Pi_{t=0}^{T_I-1} \Psi_{t+1}^{n_{t+1}^2 - n_t^2}\right)^{\frac{1}{n_{T_I}^2}}, \\
\Upsilon_2 &= -\log_{r_1} \Psi_1^{n_1}(\theta_0) - \frac{L}{\mu} \log \Psi_1^{n_1^2}(\theta_0),
\end{aligned}
$$

thus

$$
T_{a1} \leq \frac{L}{\mu} \Phi_2 + \frac{L}{\mu} n_{T_I}^2 \log \frac{\bar{\Omega}_2}{\Psi_{T_I}(\theta_{T_I})} + \Upsilon_2.
$$

With similar procedures, we have

$$
T_{a2} \leq \frac{L}{\mu} \Phi_3 + \frac{L}{\mu} n_{T_I}^3 \log \frac{\bar{\Omega}_3}{\Psi_{T_I}(\theta_{T_I})} + \Upsilon_3,
$$

where

$$\Phi_3 = \sum_{t=1}^{T_I-1} n_t^3 \log \frac{\Psi_{t+1}(\theta_t)}{\Psi_t(\theta_t)}, \quad \bar{\Omega}_3 = \left(\Pi_{t=0}^{T_I-1} \Psi_{t+1}^{n_{t+1}^3 - n_t^3}\right)^{\frac{1}{n_{T_I}^3}},$$
$$\text{and } \Upsilon_3 = -\log_{r_1} \Psi_1^{n_1^2}(\theta_0) - \frac{L}{\mu} \log \Psi_1^{n_1^3}(\theta_0).$$

5.3.2.2 Sample Screening

To remove sample i from $\mathcal{A}_t$, we need $h(x_i, y_i; w_t) - \sqrt{k_{ii} G(\theta_t)} > 0$. This leads to

$$G(\theta_t) < \frac{h^2(x_i, y_i; w_t)}{k_{ii}}.$$

Lemma 3 The time complexity for the sample screening phase is $O\Big(2\frac{L}{\mu} n_{T_I}^2 \log \frac{\Psi_{T_I+1}(\theta_{T_I})}{\bar{\Gamma}_2} + \frac{L}{K\mu} n_{T_I}^3 \log \frac{\Psi_{T_I+1}(\theta_{T_I})}{\bar{\Gamma}_3}\Big)$, where

$$\bar{\Gamma}_2 = \left(\Pi_{t=T_I+1}^{T_D-1} \Psi_t^{n_t^2 - n_{t+1}^2}(\theta_t) \Psi_{T_D}^{n_{T_D}^2}(\theta_{T_D})\right)^{\frac{1}{n_{T_I+1}^2}},$$
$$\bar{\Gamma}_3 = \left(\Pi_{t=T_I+1}^{T_D-1} \Psi_t^{n_t^3 - n_{t+1}^3}(\theta_t) \Psi_{T_D}^{n_{T_D}^3}(\theta_{T_D})\right)^{\frac{1}{n_{T_I+1}^3}}.$$

Proof: For the sample screening phase,

$$\begin{aligned} T_b &= \sum_{t=T_I+1}^{T_D} \frac{\log_{r_t} \frac{\Psi_t(\theta_t)}{\Psi_t(\theta_{t-1})}}{K} (2Kn_t + n_t^2) \\ &= 2 \sum_{t=T_I+1}^{T_D} n_t \log_{r_t} \frac{\Psi_t(\theta_t)}{\Psi_t(\theta_{t-1})} + \frac{1}{K} \sum_{t=T_I+1}^{T_D} n_t^2 \log_{r_t} \frac{\Psi_t(\theta_t)}{\Psi_t(\theta_{t-1})} \end{aligned}$$

Let

$$
\begin{aligned}
T_{b1} &= \sum_{t=T_I+1}^{T_D} n_t \log_{r_t} \frac{\Psi_t(\theta_t)}{\Psi_t(\theta_{t-1})} \leq \frac{L}{\mu} \sum_{t=T_I+1}^{T_D} n_t^2 \log \frac{\Psi_t(\theta_{t-1})}{\Psi_t(\theta_t)} \\
&= \frac{L}{\mu} \log \Pi_{t=T_I+1}^{T_D} \frac{\Psi_t^{n_t^2}(\theta_{t-1})}{\Psi_t^{n_t^2}(\theta_t)} \\
&= \frac{L}{\mu} \log \frac{\Psi_{T_I+1}^{n_{T_I+1}^2}(\theta_{T_I})}{\Pi_{t=T_I+1}^{T_D-1} \Psi_t^{n_t^2-n_{t+1}^2}(\theta_t) \Psi_{T_D}^{n_{T_D}^2}(\theta_{T_D})} \\
&= \frac{L}{\mu} \log \frac{\Psi_{T_I+1}^{n_{T_I+1}^2}(\theta_{T_I})}{\bar{\Gamma}_2^{n_{T_I+1}^2}} = \frac{L}{\mu} n_{T_I+1}^2 \log \frac{\Psi_{T_I+1}(\theta_{T_I})}{\bar{\Gamma}_2}.
\end{aligned}
$$

Here

$$
\bar{\Gamma}_2 = \left(\Pi_{t=T_I+1}^{T_D-1} \Psi_t^{n_t^2-n_{t+1}^2}(\theta_t) \Psi_{T_D}^{n_{T_D}^2}(\theta_{T_D}) \right)^{\frac{1}{n_{T_I+1}^2}}.
$$

Similarly, let

$$
T_{b2} = \sum_{t=T_I+1}^{T_D} n_t^2 \log \frac{\Psi_t(\theta_t)}{\Psi_t(\theta_{t-1})} \leq \frac{L}{\mu} n_{T_I+1}^3 \log \frac{\Psi_{T_I+1}(\theta_{T_I})}{\bar{\Gamma}_3},
$$

where

$$
\bar{\Gamma}_3 = \left(\Pi_{t=T_I+1}^{T_D-1} \Psi_t^{n_t^3-n_{t+1}^3}(\theta_t) \Psi_{T_D}^{n_{T_D}^3}(\theta_{T_D}) \right)^{\frac{1}{n_{T_I+1}^3}}.
$$

5.3.2.3 Time Cost

After the sample recruiting and screening phases, we only need to iteratively update the parameters to improve the accuracy. The time complexity for accuracy pursuing is

$$
T_c = m \log_{r_m} \frac{\varepsilon}{\Psi_{T_D}(\theta_{T_D})} \leq \frac{L}{\mu} m^2 \log \frac{\Psi_{T_D}(\theta_{T_D})}{\varepsilon}.
$$

Theorem 3 The time complexity for the proposed algorithm is $O\big(2\frac{\tau L}{\mu}n_{T_I}^2\log\frac{\bar{\Omega}_2}{\varepsilon_D}+2\frac{L}{\mu}m^2\log\frac{\varepsilon_D}{\varepsilon}+\frac{(1+\tau)L}{\mu}an_{T_I}^2\big)$. Here $\bar{\Omega}_2=\big(\Pi_{t=0}^{T_I-1}\Psi_{t+1}^{n_{t+1}^2-n_t^2}\big)^{\frac{1}{n_{T_I}^2}}$, ε_D is the minimize accuracy gap for the sample screening, m is the number of support vectors.

Proof: The time complexity for the proposed algorithm is

$$
\begin{aligned}
T &= T_a + T_b + T_c \\
&\le \frac{2K+n}{K}\Big(\frac{L}{\mu}\Phi_2 + \frac{L}{\mu}n_{T_I}^2\log\frac{\bar{\Omega}_2}{\Psi_{T_I}(\theta_{T_I})} + \Upsilon_2\Big) + \frac{1}{K}\Big(\frac{L}{\mu}\Phi_3 + \frac{L}{\mu}n_{T_I}^3\log\frac{\bar{\Omega}_3}{\Psi_{T_I}(\theta_{T_I})} + \\
&\quad \Upsilon_3\Big) + 2\frac{L}{\mu}n_{T_I+1}^2\log\frac{\Psi_{T_I+1}(\theta_{T_I})}{\bar{\Gamma}_2} + \frac{1}{K}\frac{L}{\mu}n_{T_I+1}^3\log\frac{\Psi_{T_I+1}(\theta_{T_I})}{\bar{\Gamma}_3} + 2\frac{L}{\mu}m^2\log\frac{\Psi_{T_D}(\theta_{T_D})}{\varepsilon} \\
&= 2\frac{L}{\mu}n_{T_I}^2\frac{\bar{\Omega}_2}{\bar{\Gamma}_2} + \frac{n}{K}\frac{L}{\mu}n_{T_I}^2\log\frac{\bar{\Omega}_2}{\Psi_{T_I}(\theta_{T_I})} + 2\frac{L}{\mu}m^2\log\frac{\Psi_{T_D}(\theta_{T_D})}{\varepsilon} \\
&\quad + \frac{2K+n}{K}\frac{L}{\mu}\Phi_2 + \frac{1}{K}\frac{L}{\mu}\Phi_3 + \frac{1}{K}\frac{L}{\mu}n_{T_I+1}^3\log\frac{\Psi_{T_I+1}(\theta_{T_I})}{\bar{\Gamma}_3} + \Upsilon \\
&\le 2\frac{L}{\mu}n_{T_I}^2\frac{\bar{\Omega}_2}{\Psi_{T_D}(\theta_{T_D})} + 2\frac{L}{\mu}m^2\log\frac{\Psi_{T_D}(\theta_{T_D})}{\varepsilon} + \frac{Ln_{T_I}^2}{K\mu}\log\frac{\bar{\Omega}_2^n}{\Psi_{T_I}^{n-n_{T_I}}(\theta_{T_I})\bar{\Gamma}_3^{n_{T_I}}} \\
&\quad + \frac{2K+n}{K}\frac{L}{\mu}\Phi_2 + \frac{1}{K}\frac{L}{\mu}\Phi_3 + \Upsilon \\
&\le 2\frac{L}{\mu}n_{T_I}^2\frac{\bar{\Omega}_2}{\Psi_{T_D}(\theta_{T_D})} + 2\frac{L}{\mu}m^2\log\frac{\Psi_{T_D}(\theta_{T_D})}{\varepsilon} + \frac{Ln_{T_I}^2 n}{K\mu}\log\frac{\bar{\Omega}_2}{\Psi_{T_D}(\theta_{T_D})} \\
&\quad + \frac{2K+n}{K}\frac{L}{\mu}\Phi_2 + \frac{1}{K}\frac{L}{\mu}\Phi_3 + \Upsilon \\
&= 2\frac{L}{\mu}n_{T_I}^2(1+\eta)\frac{\bar{\Omega}_2}{\Psi_{T_D}(\theta_{T_D})} + 2\frac{L}{\mu}m^2\log\frac{\Psi_{T_D}(\theta_{T_D})}{\varepsilon} + 2\frac{L}{\mu}(1+\eta)\Phi_2 + \frac{1}{K}\frac{L}{\mu}\Phi_3 + \Upsilon \\
&\le 2\frac{L}{\mu}n_{T_I}^2(1+\eta)\frac{\bar{\Omega}_2}{\Psi_{T_D}(\theta_{T_D})} + 2\frac{L}{\mu}m^2\log\frac{\Psi_{T_D}(\theta_{T_D})}{\varepsilon} + \frac{L}{\mu}(3+2\eta)\Phi_2 + \Upsilon,
\end{aligned}
$$

Here

$$\eta = \frac{n}{2K}, \text{ and } \Upsilon = \frac{2K+n}{K}\Upsilon_2 + \frac{1}{K}\Upsilon_3.$$

As

$$\Phi_2 = \sum_{t=1}^{T_I-1} n_t^2 \log \frac{\Psi_{t+1}(\theta_t)}{\Psi_t(\theta_t)},$$

we can control the value of h to ensure $\log \frac{\Psi_{t+1}(\theta_t)}{\Psi_t(\theta_t)} \leq 1$,

$$T \leq 2\frac{L}{\mu} n_{T_I}^2 (1+\eta) \frac{\bar{\Omega}_2}{\Psi_{T_D}(\theta_{T_D})} + 2\frac{L}{\mu} m^2 \log \frac{\Psi_{T_D}(\theta_{T_D})}{\varepsilon} + \frac{L}{\mu}(3+2\eta) a n_{T_I}^2 + \Upsilon.$$

Thus the time complexity is $O\big(2\frac{\tau L}{\mu} n_{T_I}^2 \frac{\bar{\Omega}_2}{\varepsilon_D} + 2\frac{L}{\mu} m^2 \log \frac{\varepsilon_D}{\varepsilon} + \frac{(1+\tau)L}{\mu} a n_{T_I}^2\big)$.

5.4 Experiments

We first compare SAIV with a typical working set (Shrinking) method [24] and then we compare SAIV with the state-of-the-art sequential sample screening method [12]. More thorough comparison studies are still ongoing, and we present the available preliminary results in this section.

5.4.1 Comparison with Shrinking Method

We evaluate the proposed method on different data sets from the LIBSVM website [86]. We compare SAIV with the shrinking method [24] and report the running time in Table 5.1. Here we use the RBF kernel for both methods. The running time for both methods are based on the same hyper-parameters (C and kernel parameters). From Table 5.1, we can see that the proposed method achieves improved computation efficiently compared with the shrinking method. Furthermore, when the data sample size is large, SAIF can reduce more computational cost.

5.4.2 Comparison with Sequential Screening

Grid search with cross validation has often been adopted to select model hyper-parameters (C and kernel parameters). In this set of experiments, we compare SAIV with the state-of-

Table 5.1: Running time (Sec.) on different data sets

Data Set	Feature Size	Sample Size	CD+Shrinking	Proposed
Gisette	5000	6000	83.8	51.1
USPS	256	7291	7.73	5.91
Vehicle	18	746	0.116	0.072

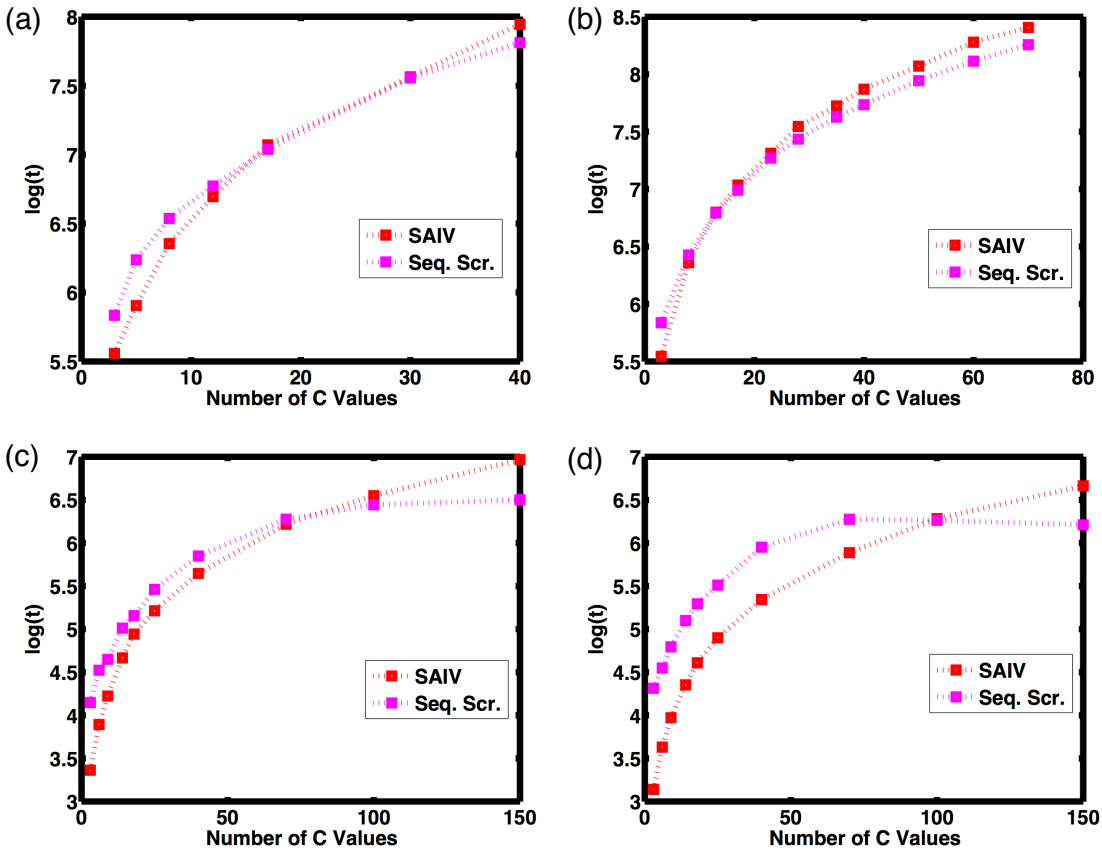


Figure 5.1: Running time for SAIV and sequential screening on Gisette (a, b) and USPS (c, d) data sets with different numbers of C values at different γ values (kernel parameter). For Gisette, γ = 1E-9 (a) and γ = 5E-8 (b). For USPS, γ =0.039 (c) and γ = 0.019 (d).

the-art sequential screening method [12] on a sequence of C values controlling the width of margin. Figure 5.1 gives the running time for both methods on Gisette and USPS data sets with different numbers of C values. For Gisette data set, all of the model hyper-parameter C values are sampled evenly on the logarithmic scale of range $[0.01, 500]$. For USPS, the range is $[0.1, 100]$. The running time for SAIV is linearly increasing with the number of C values. Although sequential screening can take less when the number of C values is large, SAIV takes less time when the number of C is small. This is because the density of C values determines the screening power of sequential screening, and smaller gaps between C values can remove more non-support samples (vectors). While the density of C values does not affect the performance of SAIV, and thus the running time for SAIV increases with the number of C values. When we do hyper-parameter tunning, we can incorporate SAIV and sequential screening with coarse to fine strategies. We can start from several different important C values with SAIV, and then do the sequential screening to select the optimal hyper-parameter.

5.5 Conclusions

In this chapter, we propose a sample selection method for SVM. The main idea is following the similar derivation of the active incremental feature selection method of SAIF for sparse learning. Theoretical analysis on convergence is given. Experiments on different data sets illustrate the advantages of the proposed method. Based on these results, we conclude that the sparse properties can reduce the model computation cost of SVM , especially when there are a large number of training samples but only a small fraction of them are support vectors. .

6. CONCLUSIONS AND FUTURE WORK

In this dissertation, we have developed several methods to scale up sparse and structure models. In Chapter 2, we present a new feature selection algorithm for LASSO, SAIF. SAIF utilizes quit different strategies compared with typical sequential and dynamic screening methods, and it actively employees the most active features and deletes inactive ones to minimize redundant computations. Experimental results prove that SAIF consumes much less computation than state of art dynamic screening method. SAIF provides a new direction for scaling up sparse learning, and it can be easily extended to group LASSO, graph LASSO, and other sparse and structure models. We also show that the idea of SAIF can be extended to support vector machines (SVM) in Chapter 5.

In Chapter 3, we try to address the GL scaling up problem. Firstly, the sequential screening rules for GL problems can be derived by formulating equivalent dual problems constrained by linear inequality systems. The bound propagation (BP) algorithm in the dual space approximates the range of sub-gradient of L_1 items, and then with the approximation we can identify as many L_1 items as possible to significantly reduce the original problem size. With dynamic screening as an efficient way to start the screening process, BP can be further improved with the the proposed transformation method. Secondly, we extend the SAIF method to GL problems with tree structures. Experimental results on both synthetic and real-world data sets demonstrate the promising performance of both methods.

We developed a scalable structured kernel feature selection in Chapter 4. With the prior knowledge of structures among features incorporated into the objective function, active regions in medical images can be robustly and efficiently identified by the proposed HSIC kernel feature selection method. The efficiency of the model can be boosted significantly

with the dual average stochastic algorithm. Experimental results on simulation data and real-world 3D image have verified the effectiveness and efficiency of the proposed method.

Based on the proposed methods in this dissertation, there are several directions we can progress further in the future. First, SAIF can be extended to more general cases, such as group LASSO, or general convex problems with sparse structures. Second, SAIF can be further improved with strategies such as multi-level active set strategies, and SGD methods [42]. Finally, kernel feature selection models can be further improved with the proposed screening methods in this dissertation. These directions can improve the model efficiency further by leveraging the sparse and structures in the data sets.

REFERENCES

[1] E. P. Xing, Q. Ho, P. Xie, and D. Wei, "Strategies and principles of distributed machine learning on big data," *Engineering*, vol. 2, 2016.

[2] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity-The Lasso and Generalizations*. CRC Press, 2015.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[4] Y.-X. Wang, V. Sadhanala, W. Dai, W. Neiswanger, S. Sra, and E. P. Xing, "Parallel and distributed block-coordinate frank-wolfe algorithms," in *ICML*, 2016.

[5] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[6] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, pp. 2121–2159, 2011.

[7] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM Journal on Control and Optimization*, vol. 30, p. 838âĂŞ855, 1992.

[8] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, and J. T. R. J. Tibshirani, "Strong rules for discarding predictors in lasso-type problems," *Journal of the Royal Statistical Society Series B*, vol. 74, pp. 245–266, 2012.

[9] J. Fan and J. Lv, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Journal of the Royal Statistical Society Series B*, vol. 70, pp. 849–911, 2008.

[10] L. E. Ghaoui, V. Viallon, and T. Rabbani, "Safe feature elimination in sparse supervised learning," *Pacific Journal of Optimization*, vol. 8, pp. 667–698, 2012.

[11] J. Wang, P. Wonka, and J. Ye, "Lasso screening rules via dual polytope projection," *arXiv.org*, 2014.

[12] K. Ogawa, Y. Suzuki, S. Suzumura, and I. Takeuchi, "Safe sample screening for support vector machines," *arXiv:1401.6740v1*, 2014.

[13] J. Wang, P. Wonka, and J. Ye, "Scaling svm and least absolute deviations via exact data reduction," *arXiv:1310.7048*, 2013.

[14] K. Ogawa, Y. Suzuki, and I. Takeuchi, "Safe screening of non-support vectors in pathwise svm computation," in *ICML*, 2013.

[15] J. Zimmert, C. S. de Witt, G. Kerg, and M. Kloft, "Safe screening for support vector machines," in *Proceedings of the NIPS 2015 Workshop on Optimization in Machine Learning (OPT)*, 2015.

[16] R. Tibshirani, "Regression shringkage and selection via the lasso," *Journal of the Royal Statistical Society Series B*, vol. 58, pp. 267–288, 1996.

[17] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society Series B*, vol. 67, pp. 91–108, 2005.

[18] W. J. Fu, "Penalized regressions: The bridge versus the lasso," *Journal of Computational and Graphical Statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[19] S. S. Chen, D. L. Donoho, and M. A. Saunders., "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1999.

[20] S. Perkins, K. Lacker, and J. Theiler, "Grafting: Fast, incremental feature selection by gradient descent in function space," *Journal of Machine Learning Research*, vol. 3, pp. 1333–1356, 2003.

[21] J. Wang, J. Zhou, J. Liu, P. Wonka, and J. Ye, "A safe screening rule for sparse logistic regression," in *Advances in Neural Information Processing Systems*, 2014.

[22] J. Wang, PeterWonka, and J. Ye, "Scaling svm and least absolute deviations via exact data reduction," in *ICML*, 2014.

[23] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "Coordinate descent method for large-scale l2-loss linear support vector machines," *Journal of Machine Learning Research*, vol. 9, pp. 1369–1398, 2008.

[24] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear svm," in *ICML*, 2008.

[25] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using second order information for training support vector machines," *Journal of Machine Learning Research*, vol. 6, p. 1889âĂŞ1918, 2005.

[26] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods - Support Vector Learning*, MIT Press, 1998.

[27] J. Kivinen, A. J. Smola, and R. C. Williamson., "Online learning with kernels," *IEEE Transactions on Signal Processing*, vol. 52, p. 8, 2004.

[28] A. Bordes, L. Bottou, and P. Gallinari, "Sgd-qn: Careful quasi-newton stochastic gradient descent," *Journal of Machine Learning Research*, vol. 10, pp. 1737–1754, 2009.

[29] M. Masaeli, G. Fung, and J. G. Dy, "From transformation-based dimensionality reduction to feature selection," in *ICML*, 2010.

[30] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer Press, 2003.

[31] M. Yamada, W. Jitkrittum, L. Sigal, E. P. Xing, and M. Sugiyama, "High-dimensional feature selection by feature-wise kernelized lasso," *Neural Computation*, vol. 26, pp. 185–207, 2014.

[32] F. Li, Y. Yang, and E. P. Xing, "From lasso regression to feature vector machine," in *NIPS*, 2006.

[33] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research*, vol. 13, pp. 795–828, 2012.

[34] R. J. Tibshirani and J. Taylor, "The solution path of the generalized lasso," *The Annals of Statistics*, vol. 39, no. 3, pp. 1335–1371, 2011.

[35] J. Wang, W. Fan, and J. Ye, "Fused lasso screening rules via the monotonicity of subdifferentials," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, 2015.

[36] J. Wang and J. Ye, "Two-layer feature reduction for sparse-group lasso via decomposition of convex sets," in *Advances in Neural Information Processing Systems*, 2014.

[37] J. Wang and J. Ye, "Multi-layer feature reduction for tree structured group lasso via hierarchical projection," in *Advances in Neural Information Processing Systems*, 2015.

[38] E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon, "Gap safe screening rules for sparse multi-task and multi-class models," in *NIPS*, 2015.

[39] O. Fercoq, A. Gramfort, and J. Salmon, "Mind the duality gap: safer rules for the lasso," in *ICML*, 2015.

[40] E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon, "Dynamic screening: Accelerating first-order algorithms for the lasso and group-lasso," *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 63, 2015.

[41] Z. Zhao, J. Liu, and J. Cox, "Safe and efficient screening for sparse support vector machine," in *KDD*, 2014.

[42] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *Journal of Machine Learning Research*, pp. 2543–2596, 2010.

[43] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[44] M. R. Osborne, B. Presnell, and B. Turlach, "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis*, vol. 20, p. 389âĂŞ403, 2000.

[45] D. Malioutov, M. Cetin, and A. Willsky, "Homotopy continuation for sparse signal representation," in *ICASSP*, 2005.

[46] P. J. Garrigues and L. E. Ghaoui, "Homotopy continuation for sparse signal representation," in *NIPS*, 2008.

[47] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010.

[48] T. Zhao, H. Liu, and T. Zhang, "Pathwise coordinate optimization for sparse learning: Algorithm and theory," *arXiv*, 2017.

[49] T. B. Johnson and C. Guestrin, "Blitz: A principled meta-algorithm for scaling sparse optimization," in *International Conference on Machine Learning*, 2015.

[50] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM J. OPTIM.*, vol. 22, pp. 341–362, 2012.

[51] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type method," *SIAM J. OPTIM.*, vol. 23, p. 2037âĂŞ2060, 2013.

[52] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, p. 3âĂŞ34, 2015.

[53] X. Li, T. Zhao, R. Arora, H. Liu, and M. Hong, "On faster convergence of cyclic block coordinate descent-type methods for strongly convex minimization," *arXiv*, 2017.

[54] H.-Y. Chuang, E. Lee, Y.-T. Liu, D. Lee, and T. Ideker, "Network-based classification of breast cancer metastasis," *Molecular Systems Biology*, vol. 3, p. 140, 2007.

[55] C.-C. Chang and C.-J. Lin, "Libsvm : a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 27, pp. 1–27, 2011.

[56] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society Series B*, vol. 68, pp. 49–67, 2006.

[57] B. Xin, Y. Kawahara, Y. Wang, and W. Gao, "Efficient generalized fused lasso with its application to the diagnosis of alzheimers disease," in *AAAI*, 2014.

[58] S. Ren, S. Huang, J. A. Onofrey, X. Papademetris, and X. Qian, "A scalable algorithm for structured kernel feature selection," in *AIStats*, 2015.

[59] S. Ren and X. Qian, "Structured sparse pca to identify mirna co-regulatory modules," in *ICASSP*, 2014.

[60] T. B. Arnold and R. J. Tibshirani, "Efficient implementations of the generalized lasso dual path algorithm," *Journal of Computational and Graphical Statistics*, vol. 25, no. 1, pp. 1–27, 2016.

[61] "genlasso." https://github.com/statsmaths/genlasso.

[62] "glmgen." https://github.com/statsmaths/glmgen.

[63] "SLEP:Sparse Learning with Efficient Projections." http://yelab.net/software/SLEP/.

[64] "MALSAR: Multi-task Learning via Structural Regularization." http://www.yelab.net/software/MALSAR/.

[65] Z. J. Xiang and P. J. Ramadge, "Fast lasso screening tests based on correlations," in *ICASSP*, 2012.

[66] Y. Nesterov, "Introductory lectures on convex optimization," *Applied Optimization*, vol. 87, 2004.

[67] J. Liu, Z. Zhao, J. Wang, and J. Ye, "Safe screening with variational inequalities and its application to lasso," in *ICML*, 2014.

[68] K. Korovin and A. Voronkov, "Solving systems of linear inequalities by bound propagation," *Automated Deduction-CADE-23*, vol. 6803, 2011.

[69] "IBM CPLEX Optimizer." http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[70] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Sure independence screening for ultrahigh dimensional feature spaces," *Foundations and Trends in Machine Learning*, vol. 3, pp. 1–122, 2010.

[71] "CVX: Matlab Software for Disciplined Convex Programming." http://cvxr.com/cvx/.

[72] S. Yang, L. Yuan, Y.-C. Lai, X. Shen, P. Wonka, and J. Ye, "Feature grouping and selection over an undirected graph," in *KDD*, 2012.

[73] "Alzheimer's Disease Neuroimaging Initiative (ADNI) database." http://adni.loni.usc.edu/.

[74] S. Huang, J. Li, L. Sun, J. Liu, T. Wu, K. Chen, A. Fleisher, E. Reiman, and J. Ye, "Learning brain connectivity of alzheimer's disease from neuroimaging data," in *NIPS*, 2012.

[75] I. Rodriguez-Lujan, R. Huerta, C. Elkan, and C. S. Cruz, "Quadratic programming feature selection," *Journal of Machine Learning Research*, vol. 11, pp. 1491–1516, 2010.

[76] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt, "Feature selection via dependence maximization," *Journal of Machine Learning Research*, vol. 13, pp. 1393–1434, 2012.

[77] A. Gretton, O. Bousquet, A. Smola, and B. Schlkop, "Measuring statistical dependence with hilbert-schmidt norms," *Algorithmic Learning Theory*, 2005.

[78] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman, "Sparse additive models," *Journal of Machine Learning Research*, vol. 71, pp. 1009–1030, 2009.

[79] R. Tomioka, T. Suzuki, and M. Sugiyama, "Super-linear convergence of dual augmented lagrangian algorithm for sparsity regularized estimation," *Journal of Machine Learning Research*, vol. 12, pp. 1537–1586, 2011.

[80] H. Yang, Z. Xu, I. King, and M. R. Lyu, "Online learning for group lasso," in *ICML*, 2010.

[81] B. B. Biswal, M. Mennes, X.-N. Zuo, S. Gohel, C. Kelly, S. M. Smith, C. F. Beckmann, J. S. Adelstein, R. L. Buckner, S. Colcombe, A.-M. Dogonowski, M. Ernst, D. Fair, M. Hampson, M. J. Hoptman, J. S. Hyde, V. J. Kiviniemi, R. Kötter, S.-J. Li, C.-P. Lin, M. J. Lowe, C. Mackay, D. J. Madden, K. H. Madsen, D. S. Margulies, H. S. Mayberg, K. McMahon, C. S. Monk, S. H. Mostofsky, B. J. Nagel, J. J. Pekar, S. J. Peltier, S. E. Petersen, V. Riedl, S. A. R. B. Rombouts, B. Rypma, B. L.

Schlaggar, S. Schmidt, R. D. Seidler, G. J. Siegle, C. Sorg, G.-J. Teng, J. Veijola, A. Villringer, M. Walter, L. Wang, X.-C. Weng, S. Whitfield-Gabrieli, P. Williamson, C. Windischberger, Y.-F. Zang, H.-Y. Zhang, F. X. Castellanos, and M. P. Milham, "Toward discovery science of human brain function," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4734–4739, 2010.

[82] D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes, "Nonrigid registration using free-form deformations: Application to breast MR images," *Medical Imaging, IEEE Transactions on*, vol. 18, pp. 712–721, aug. 1999.

[83] K. J. Friston, J. Ashburner, C. D. Frith, J.-B. Poline, J. D. Heather, and R. S. J. Frackowiak, "Spatial registration and normalization of images," *Human Brain Mapping*, pp. 165–189, 1995.

[84] V. Fonov, A. Evans, K. Botteron, C. Almli, R. McKinstry, D. Collins, and Brain Development Cooperative Group, "Unbiased average age-appropriate atlases for pediatric studies," *NeuroImage*, vol. 54, no. 1, pp. 317–323, 2011.

[85] J. Nutini, M. Schmidt, I. H. Laradji, M. Friedlander, and H. Koepke, "Coordinate descent converges faster with the gauss-southwell rule than random selection," *arXiv:1506.00552*, 2015.

[86] "LIBSVM Data: Classification, Regression, and Multi-label." https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/.